

AD-A041 319

ROME AIR DEVELOPMENT CENTER GRIFFISS AFB N Y

F/G 9/1

FAA LIGHTNING PROTECTION STUDY: LIGHTNING INDUCED TRANSIENTS ON--ETC(U)

MAY 77 J D NORDGARD, C CHEN

DOT-FA72WAI-353

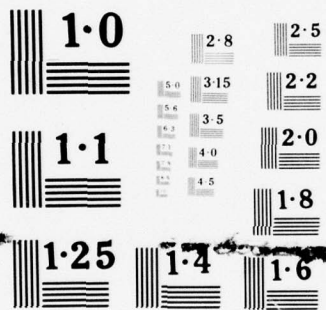
UNCLASSIFIED

FAA/RD-77-83

NL

1 OF 2  
ADA  
.041319





NATIONAL BUREAU OF STANDARDS  
MICROCOPY RESOLUTION TEST CHART

AD A 041319

Report No. FAA-RD-77-83

FAA LIGHTNING PROTECTION STUDY:  
LIGHTNING - INDUCED TRANSIENTS ON  
BURIED SHIELDED TRANSMISSION LINES:  
NUMERICAL ANALYSIS AND RESULTS

John D. Nordgard

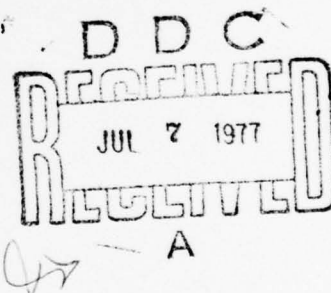
Chin-Lin Chen



COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION

May 1977

Final Report



Document is available to the U.S. public through  
the National Technical Information Service,  
Springfield, Virginia 22161.

AD No. \_\_\_\_\_  
DDC FILE COPY.

Prepared for  
**U.S. DEPARTMENT OF TRANSPORTATION**  
**FEDERAL AVIATION ADMINISTRATION**  
**Systems Research & Development Service**  
**Washington, D.C. 20590**

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

Technical Report Documentation Page

1. Report No. <b>FAA/RD-77-83</b>	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle <b>FAA Lightning Protection Study: Lightning Induced Transients on Buried Shielded Cables; Numerical Analysis and Results</b>	5. Report Date <b>May 1977</b>	6. Performing Organization Code <b>RADC/RBC</b>
7. Author(s) <b>John D. Nordgard and Chin-Lin Chen</b>	8. Performing Organization Report No.	
9. Performing Organization Name and Address <b>Department of Defense U.S. Air Force Rome Air Development Center Rome, New York 13441</b>	10. Work Unit No. (TRAIS) <b>9567 0006</b>	
12. Sponsoring Agency Name and Address <b>Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington, D.C. 20590</b>	11. Contractor or Grant No. <b>DOT-FA-72WAI-353</b>	
15. Supplementary Notes  <b>Prepared by Post Doctoral Program Rome Air Development Center</b>	13. Type of Report and Period Covered <b>Final Report, May 1976-May 1977</b>	
14. Sponsoring Agency Code <b>ARD-350</b>		
16. Abstract This report is primarily concerned with the analysis of induced transient current and voltage pulses on buried shielded transmission lines, due to earth conduction effects of nearby lightning discharges. A numerical method is presented in this report to determine the amount of coupling between a lightning discharge to ground and an earth-return transmission line. The transmission line is assumed to be a long straight horizontal coaxial cable with an inner shield and an outer armor, terminated on both ends with typical communication equipment load impedances. The general case is considered here, in which the outermost conductor is not necessarily in perfect contact with the conducting earth but has a contact impedance with the earth, as in cables with an outer dielectric covering for corrosion or water protection. Indirect strikes to the cable via conductive coupling mechanisms through the earth are considered. Several average lightning channel conditions and several representative buried cable geometries are examined. The results are conveniently displayed via numerous graphs of the time histories and frequency spectra of the resulting transient current and voltage surges.		
17. Key Words  <b>Lightning, transmission lines, induced transients</b>	18. Distribution Statement Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151	
19. Security Classif. (of this report)  <b>UNCLASSIFIED</b>	20. Security Classif. (of this page)  <b>UNCLASSIFIED</b>	21. No. of Pages  <b>119</b>
		22. Price

## Table of Contents

AUTHORITY DATE BY DISTRIBUTION AVAILABILITY CODES FILE #	23 874
--	-----------

Foreword .....	vi
Biographies.....	vii
Acknowledgement.....	ix
CHAPTER 1 INTRODUCTION.....	1
1.1 FAA Lightning Protection Study.....	1
1.2 Problem Statement.....	2
1.3 Overview.....	3
CHAPTER 2 PROBLEM GEOMETRY AND PULSE MODEL.....	5
2.1 Geometry.....	
2.2 Pulse Model.....	
CHAPTER 3 DISCUSSION OF THEORETICAL RESULTS TO BE PROGRAMMED.....	8
3.1 Vertical Dipole (Free Space).....	8
3.2 Vertical Dipole (Above Flat Earth).....	9
3.3 Ground Stroke.....	10
3.4 Governing Equations.....	11
3.4.1 Faraday's Law.....	11
3.4.2 Ampere's Law.....	12
3.4.3 Transmission Line Model.....	12
3.5 Natural Modes.....	13
3.6 Green's Functions.....	17
3.7 Transfer Functions.....	19
CHAPTER 4 THE COMPUTER PROGRAM.....	21
4.1 Introduction.....	21
4.2 Glossary of the Routines.....	21
4.3 The Input Routine.....	22
CHAPTER 5 NUMERICAL RESULTS.....	23
CHAPTER 6 CONCLUSIONS.....	26
REFERENCES.....	27
APPENDIX A Program Listing.....	
APPENDIX B Sample of Input Data Listings.....	
APPENDIX C FAA Lightning Study Participant.....	
Table 1 List of Input Cards and Formats.....	
Table 2 Peak current and voltage, rise time and half-life decay time of pulses on the inner conductor.....	

## FOREWORD

The Post-Doctoral Program at Rome Air Development Center is pursued via Project 9567 under the direction of Dr. W. W. Everett, Jr. and Mr. J. Scherer. The Post-Doctoral Program is a cooperative venture between RADC and the participating universities: Syracuse University (Department of Electrical and Computer Engineering), the U. S. Air Force Academy (Department of Electrical Engineering), Purdue University (School of Electrical Engineering), University of Kentucky (Department of Electrical Engineering), Georgia Institute of Technology (School of Electrical Engineering), Clarkson College of Technology (Department of Electrical Engineering), State University of New York at Buffalo (Department of Electrical Engineering), Florida Technological University (Department of Electrical Engineering), Florida Institute of Technology (College of Engineering), Air Force Institute of Technology (Department of Electrical Engineering), and the University of Adelaide (Department of Electrical Engineering), in South Australia. The Post-Doctoral Program provides, via contract, the opportunity for faculty and visiting faculty at the participating universities to spend a year full time on exploratory development and operational problem-solving efforts with the postdoctorals splitting their time between RADC (or the ultimate customer) and the educational institutions.

This effort was conducted via RADC Job Order No. 9567 0006 for the Federal Aviation Administration. Mr. Fred Sakate was the FAA focal point and participated closely in the technical coordination meetings and cable testing sessions.

## BIOGRAPHIES

John D. Nordgard is an Associate Professor in the School of Electrical Engineering, Georgia Institute of Technology. He received the B.E.E. (1966) from Georgia Institute of Technology and the M.S. (1967) and PH.D. (1969) degrees in Applied Physics from California Institute of Technology. He has worked as Electronic Engineer for the Charleston Naval Shipyard and as Associate Engineer for the Jet Propulsion Laboratory. He spent one year at the University of Oslo doing research on the polar ionosphere. He joined the Georgia Tech faculty in 1970 where he is engaged in teaching electromagnetic field theory and doing research in related areas. His research interest includes the areas of moving media, heterogeneous media, plasma physics, scattering, high field emission, illumination, and multiconductor transmission media. His memberships include Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi, Sigma Xi, IEEE, and IES.

Chin-Lin Chen was born in Honan, China, on March 27, 1937. He received the B.S.E.E. degree from National Taiwan University, Taipei, Taiwan, China, in 1958, the M.S. degree from North Dakota State University, Fargo, ND, in 1961, and the Ph.D. degree from Harvard University, Cambridge, MA, in 1965.

He served as a teaching assistant at North Dakota State University from 1959 to 1961 and as a research assistant and Teaching Fellow at Harvard University from 1962 to 1965. From 1965 to 1966, he was a Research Fellow in the Division of Engineering and Applied Physics, Harvard University. In 1966, he joined the School of Electrical Engineering, Purdue University, West Lafayette, IN, as an assistant professor. He is now an associate professor of Electrical Engineering. His research interests are in the areas of scattering and diffraction of electromagnetic waves, antenna radiation, surface acoustic wave devices, ultrasonics, transient protection devices, and the application of interactive computer graphics in electrical engineering education.

Dr. Chen is a member of Sigma Xi, Eta Kappa Nu, and IEEE and the American Society for Engineering Education. Since 1966, he has been a reviewer for the IEEE Transactions on Antennas and Propagations.

#### ACKNOWLEDGEMENT

The authors wish to acknowledge the contributions of Mr. Fred Sakate of the FAA, Dr. W. W. Everett, Jr. and Mr. J. Scherer of the RADC Post-Doctoral Program in formulating the program approach and for their guidance throughout the course of this study. They are indebted to Dr. B. J. Leon and Mr. Warren Peele of Purdue University for their support and encouragement. To all other members of the Post-Doctoral Program team on the FAA Lightning Protection Study, and in particular to Mr. R. M. Cosel (F. I. T.), Mr. Keith Huddleston (Georgia Tech), Dr. R. Larson (Georgia Tech), and Dr. A. W. Revay, Jr. (F. I. T.), the authors would like to express their appreciation for their contributions and assistance.

## CHAPTER 1

### Introduction

The work described in this report is part of a larger study program to provide protection for communication electronics equipment against transient electromagnetic disturbances. Strong electromagnetic disturbances may be caused by nearby lightning activities or man-made electromagnetic pulses. Current and voltage pulses are induced in nearby cables running between buildings or equipment enclosures. These currents and voltages are then coupled into the terminal equipment. Excessive interference in conventional exposed metallic communication lines is indicated unless adequate protection measures are provided. This may require extra electromagnetic shielding of certain important communication lines and associated buildings housing sensitive equipment and/or the installation of protective devices on certain communication equipment.

The larger study, known as the FAA Lightning Protection Study, has been performed by the Post-Doctoral Program through several of its member universities for the Federal Aviation Administration. The institutions include the Air Force Institute of Technology, Florida Institute of Technology, Georgia Institute of Technology, and Purdue University. The individual participants in the FAA Lightning Protection Study are listed in Appendix C.

#### 1.1 FAA Lightning Protection Study

The FAA Lightning Protection Study consists of three technical tasks: (1) the determination of the voltage and current levels likely to be conducted to FAA equipment; (2) the determination of the susceptibility levels of FAA equipment; and (3) the determination of lightning protective devices that are available to reduce the levels of (1) to those permitted by (2).

These three tasks have been performed in parallel with close interaction and are essentially completed. (1), (2), (3), (4) Appendix C lists the schools having primary responsibility for each of the tasks.

This report is the result of the work done under the first technical task. This study is primarily concerned with the numerical analysis of the resultant circuit disturbances caused by earth conduction effects of lightning discharges. A numerical model is presented in this study to determine the amount of coupling between a lightning discharge and an earth-return transmission line. Only the detailed numerical calculations are to be reported in this report since the theoretical foundation and analysis were presented in an earlier report. (4) Actually, this work can be considered as an extension of that earlier report since the earlier report only contained a description of the theory of lightning-induced transients; whereas, this present report contains the applications of that theory to a typical buried shielded transmission line.

## 1.2 Problem Statement

Numerous interference and protection problems are encountered in the development and operation of extensive communication and power systems. These problems are caused by the internal coupling of such systems with each other and by the external presence of the earth which, in some measure, is involved as a return conductor. The earth also serves as a return conductor for lightning currents, which often occasion disturbances in communication and power circuits. Lightning disturbances are largely atmospheric phenomena governed by the physical properties of the air. However, the behavior and effects of the lightning near the surface of the ground in communication and power systems are primarily earth conduction problems caused by the finite conductivity of the earth. Therefore, problems arise both in communication and power system circuits concerning the protection of transmission lines and

associated equipment against interference and possible breakdown caused by excessive voltage or current surges induced by lightning discharges.

To deal adequately with such problems, it is necessary to consider solutions to the basic problem in which the earth, as well as conducting current paths, are involved in the lightning discharge. The analysis of such problems is inherently more complicated than the problem of completely metallic circuits embedded in an insulating medium, since the great extent of the earth necessitates the use of electromagnetic field theory, rather than conventional transmission line or circuit theory, in the solution of most aspects of the problem. It is, therefore, necessary to restrict the analysis to fairly simple fundamental cases, in which simplified models of the earth, cable, and lightning channel geometries are used, on account of the complexities that would otherwise arise. Therefore, ionization effects caused by high induced voltages or electrolytic actions are not considered. Also, the heterogeneous character of the earth as a conductor and an electrolyte are not considered. Furthermore, the extremely variable nature of the lightning currents and voltages are not considered; however, typically average values of the lightning channel parameters are used.

### 1.3 Overview

This report is organized as follows.

First, the geometry of the cable and the lightning channel are stated in Chapter 2. Then, the basic equations which govern the behavior of the electromagnetic disturbances caused by earth conduction effects of lightning discharges are summarized in Chapter 3. These equations were derived in the earlier theoretical analysis <sup>(4)</sup> and are to be studied and programmed in this present numerical analysis. These equations include: the fields due to an electric dipole in free space; the fields of a vertically or horizontally

oriented dipole above a flat earth; the mutual coupling impedances between the dipole-like lightning channel and a buried wire; the equivalent distributed transmission line formulas of the coupling phenomena; the induced electric field intensity along the outer conductor of the cable for a lightning stroke to ground; the Green's Functions for a distributed voltage or current source; characteristic equation of the transmission line; the characteristic values of the transmission line, e.g., the propagation constant and associated propagation modes; the induced current and voltage surges on the outer armor of the cable; the voltage and current standing waves on the transmission line; the induced current and voltage surges on the center conductor of the coaxial cable; and the impedance transfer functions of an armored and shielded coaxial cable. The details of the above theoretical analysis were presented in the earlier report. <sup>(4)</sup> The numerical techniques used to compute the theoretical formulas given above are discussed in Chapter 4. A listing of the resulting computer code is given in Appendix A. A typical sample of the input data is given in Appendix B.

Finally, the output results are displayed in Chapter 5 via numerous graphs of the time histories and frequency spectra of the resulting transient current and voltage surges. Some intermediate results are also presented.

## CHAPTER 2

### Problem Geometry and Pulse Model

#### 2.1 Geometry

The geometry of the cable and the lightning channel are shown in Figure 1.

The cable is buried at a uniform depth  $d$  below the surface of the earth. The cable is coaxial with an inner core, an intermediate shield and armor, and an outer sheath for corrosion and mechanical protection. The cable has a length  $\ell$  and is terminated in typical load impedances  $Z_{\ell}^{\pm}$  at each end, which are grounded to the armor and the shield.

A ground stroke of lightning carrying a total current  $I$  terminates at a distance  $y_s$  from the cable. The lightning channel is vertical and extends to a height  $h$  above the ground. The lightning channel carries a typical "double exponential" return stroke current pulse, with a rise time on the order of 1  $\mu$ s and a decay time on the order of 50  $\mu$ s. The peak current of a typical return stroke is on the order of 20 KA.

#### 2.2 Pulse Model

The current pulses produced in a lightning stroke may be closely represented by a "double exponential" distribution

$$I(t) = I_0(e^{-\alpha t} - e^{-\beta t}) \quad (1)$$

The pulse rises rapidly to a peak value  $I_p$  at the time  $t_p$  and decays slowly to a value of  $cI_p$  at the time  $t_d$ , where  $c < 1$ ,

$$I_p = I(t_p) = I_0(e^{-\alpha t_p} - e^{-\beta t_p}) \quad (2)$$

$$cI_p = I(t_d) = I_0(e^{-\alpha t_d} - e^{-\beta t_d}) \quad (3)$$

A relation between  $\alpha$ ,  $\beta$ , and  $t_p$  can be determined from the expression for the extremum

$$\left. \frac{dI(t)}{dt} \right|_{t=t_p} = 0, \quad (4)$$

which implies that

$$\alpha/\beta = e^{(\alpha-\beta)t_p} \quad (5)$$

or

$$\alpha e^{-\alpha t_p} = \beta e^{-\beta t_p} \quad (6)$$

A relation between  $\alpha$ ,  $\beta$ ,  $t_p$ , and  $t_d$  may be obtained from (2) and (3),

$$(e^{-\alpha t_d} - e^{-\beta t_d}) = c(e^{-\alpha t_p} - e^{-\beta t_p}) \quad (7)$$

$\alpha$  and  $\beta$  may be solved numerically from (5) and (7) once  $c$ ,  $t_p$ , and  $t_d$  are specified. A simple and accurate approximation may be made by noting that  $t_d \gg t_p$ . Then (3) may be approximated by

$$cI_p \approx I_0 e^{-\alpha t_d} \quad (8)$$

Combination of (2) and (3), yields

$$I_p = I_0 \left(1 - \frac{\alpha}{\beta}\right) e^{-\alpha t_p} \quad (9)$$

An expression for  $\alpha$  may be obtained from (8) and (9),

$$\alpha = \frac{\ln c + \ln(1 - \frac{\alpha}{\beta})}{t_p - t_d} \quad (10)$$

When  $t_d \gg t_p$ ,  $\alpha \ll \beta$ , the above equation may be further simplified,

$$\alpha = (\ln c)/(t_p - t_d) \quad (11)$$

Knowing  $\alpha$ , a numerical value for  $\beta$  may be calculated from (6) using, for example, a Newton-Raphson technique.

Knowing the complete wave form  $I(t)$  in the time-domain, the frequency spectrum  $i(\omega)$  of the current in the frequency-domain is determined by

$$i(\omega) = I_0 \left( \frac{1}{\alpha - j\omega} - \frac{1}{\beta - j\omega} \right). \quad (12)$$

For a typical lightning stroke,

$$t_r = 1.2 \mu s$$

$$t_d = 50 \mu s$$

and

$$I_0 = 20 \text{ KA}$$

Using the above parameters, the resulting wave for  $I(t)$  in the time-domain is shown graphically in Figure 2; the resulting wave spectrum  $i(\omega)$  in the frequency-domain is shown graphically in Figure 3.

The computer program superposes pulse trains of the above "double exponential" form to model the prestrikes, the return stroke, and any subsequent restrikes.

## CHAPTER 3

### Discussion of Theoretical Results to be Programmed

The theoretical results of the earlier report <sup>(4)</sup> are now summarized and discussed. Detailed description of symbols and definitions of terms can be found there.

#### 3.1 Vertical Dipole (Free Space)

The current density  $\underline{J}$  of a vertically polarized point dipole with a total current  $I$  at a distance  $z_0$  on the  $z$  axis in free space, as shown in Figure 4, is approximated by

$$\underline{J} = \hat{z} I d\ell \delta(z-z_0) . \quad (13)$$

Then, the resulting Hertz potential  $\underline{\Pi}$  is

$$\underline{\Pi} = \hat{z} j \frac{I d\ell}{4\pi\omega\epsilon_0} \frac{e^{-jk_0 R}}{R} \quad (14)$$

where the free space wave number  $k_0$  is given by

$$k_0 = \omega \sqrt{\mu_0 \epsilon_0}$$

and the distance  $R$  is given by

$$R = \sqrt{\rho^2 + (z-z_0)^2} .$$

Alternatively, by applying a cylindrical Hankel transformation,

$$\underline{\Pi} = \hat{z} j \frac{I d\ell}{4\pi\omega\epsilon_0} \int_0^\infty dt \frac{\tau}{\epsilon_0} J_0(\tau \rho) e^{-\xi_0 |z-z_0|} \quad (15)$$

where  $J_0$  is the Bessel function of order zero and where the axial wave number  $\xi_0$  is defined by

$$\xi_0 = \sqrt{\tau^2 - k_0^2}$$

Notice that there is no  $\phi$  dependence due to the axial symmetry of the geometry.

### 3.2 Vertical Dipole (Above Flat Earth)

The Hertz potentials of a vertically polarized point dipole with a total current  $I$  a distance  $z_0$  on the  $z$  axis above a flat earth, as shown in Figure 5, have the form

$$\Pi_a = \hat{z} j \frac{Id\ell}{4\pi\omega\epsilon_0} \int_0^\infty d\tau J_0(\tau\rho) \left( \frac{\tau}{\xi_a} e^{-\xi_a|z-z_0|} + r(\tau)e^{-\xi_a z} \right), (z>0) \quad (17)$$

$$\Pi_e = \hat{z} j \frac{Id\ell}{4\pi\omega\epsilon_0} \int_0^\infty d\tau J_0(\tau\rho) t(\tau)e^{+\xi_e z} \quad (z<0) \quad (18)$$

where  $\Pi_a$  and  $\Pi_e$  are, respectively, the potentials in the air and in the earth. Also,  $r$  is the reflection coefficient of the wave reflected from the air-earth interface and  $t$  is the transmission coefficient of the wave transmitted through the air-earth interface. From the boundary conditions on the fields at the air-earth interface,

$$r(\tau) = \frac{\tau}{\xi_a} \left( 1 - 2 \frac{k_a^2 \xi_e}{k_e^2 \xi_a + k_a^2 \xi_e} \right) e^{-\xi_a z_0} \quad (19)$$

$$t(\tau) = 2\tau \frac{k_a^2}{k_e^2 \xi_a + k_a^2 \xi_e} e^{-\xi_a z_0} \quad (20)$$

where the axial wave numbers  $\xi_a$  and  $\xi_e$  are defined by

$$\xi_a = \sqrt{\tau^2 - k_a^2}$$

$$\xi_e = \sqrt{\tau^2 - k_e^2}$$

and where the wave numbers  $k_a$  and  $k_e$  are defined by

$$k_a = k_0 = \omega\sqrt{\mu_0\epsilon_0}$$

$$k_e = \omega\sqrt{\mu_e(\epsilon_e + j\frac{\sigma_e}{\omega})}$$

Alternately, by applying a cylindrical Hankel transformation,

$$\underline{\Pi}_a = \hat{z} j \frac{Id\ell}{4\pi\omega\epsilon_0} \left( \frac{e^{-jk_a R}}{R} + \frac{e^{-jk_a R'}}{R'} - \Lambda_a \right) \quad (21)$$

$$\underline{\Pi}_e = \hat{z} j \frac{Id\ell}{4\pi\omega\epsilon_0} \Lambda_e \quad (22)$$

where the distances  $R$  and  $R'$  are defined by

$$R = \sqrt{\rho^2 + (z-z_0)^2}$$

$$R' = \sqrt{\rho^2 + (z+z_0)^2}$$

and the terms  $\Lambda_a$  and  $\Lambda_e$  are defined by

$$\Lambda_a = 2k_a^2 \int_0^\infty d\tau \tau J_0(\tau\rho) \frac{\frac{\xi_e}{\xi_a}}{k_e^2 \xi_a + k_a^2 \xi_e} e^{-\xi_a(z+z_0)} \quad (23)$$

$$\Lambda_e = 2k_a^2 \int_0^\infty d\tau \tau J_0(\tau\rho) \frac{1}{k_e^2 \xi_a + k_a^2 \xi_e} e^{+\xi_e z} e^{-\xi_a z_0} \quad (24)$$

The first two terms involving  $R$  and  $R'$  in (21) for  $\underline{\Pi}_a$  represent the fields due to the dipole and its image through a perfectly conducting earth. The terms  $\Lambda_a$  and  $\Lambda_e$  in (21) and (22) for  $\underline{\Pi}_a$  and  $\underline{\Pi}_e$  are, respectively, the correction terms in the air and in the earth, which account for the finite conductivity of the earth.

### 3.3 Ground Stroke

The Hertz potential  $\underline{\Pi}_s$  in the earth due to a vertical lightning stroke to ground with a total height  $h$  is determined by integrating the Hertz potential  $\underline{\Pi}_e$  in the earth along the  $z$  axis from 0 to  $h$ , i.e.,

$$\Pi_s = \int_0^h dz' \Pi_e = \hat{z} j \frac{I}{4\pi\omega\epsilon_0} \int_0^\infty dz' \int_0^\infty d\tau 2\tau J_0(\tau\rho) \frac{k_a^2}{k_e^2 \xi_a + k_a^2 \xi_e} e^{+\xi_e z - \xi_a z'} \quad (25)$$

In the above expression, it is assumed that  $h \gg 1$ .

The corresponding electrostatic potential  $\phi_s$  in the earth is

$$\phi_s = \phi_e = -j \frac{I}{2\pi\omega\epsilon_0} \int_0^\infty d\tau \tau J_0(\tau\rho) \frac{k_a^2 \frac{\xi_e}{\xi_a}}{k_e^2 \xi_a + k_a^2 \xi_e} e^{+\xi_e z}, \quad (z < 0) \quad (26)$$

and the corresponding field intensity  $E_s$  in the earth along the position of the buried cable is

$$E_s = E_e \Big|_{\substack{y=y_s \\ z=-d}} = -j \frac{I}{2\pi\omega\epsilon_0} \frac{x}{R_s^3} \int_0^\infty d\tau \tau^2 J_1(\tau\rho) \frac{k_a^2 \frac{\xi_e}{\xi_a}}{k_e^2 \xi_a + k_a^2 \xi_e} e^{-\xi_e d}, \quad (z < 0) \quad (27)$$

where

$$R_s = \sqrt{x^2 + y_s^2}$$

### 3.4 Governing Equations

#### 3.4.1 Faraday's Law

When Faraday's Law

$$\oint_{C_1} \frac{d\mathbf{l}}{C_1} \cdot \mathbf{E} = -j\omega \int_{S_1} \frac{d\mathbf{S}}{S_1} \cdot \mathbf{B} \quad (28)$$

is applied to the contour  $C_1$  and the surface  $S_1$ , as shown in Figure 6, the rate of change of the voltage wave  $V$  along the cable is

$$\frac{dV(x)}{dx} = -j\omega\phi(x) - E(x) \quad (29)$$

where the voltage  $V$  and the potential  $\phi$ , per unit length in the  $x$ -direction, are defined by

$$V(x) = - \int_{\infty}^{r_i} dr E_{re} - \int_{r_i}^{r_{\omega}} dr E_{ri} \quad (30)$$

$$\phi(x) = - \int_{\infty}^{r_i} dr B_{\theta e} - \int_{r_i}^{r_{\omega}} dr B_{\theta i} \quad (31)$$

#### 3.4.2 Amperes' Law

When Amperes' Law

$$\oint_{C_2} \underline{dl} \cdot \underline{H} = j\omega \int_{S_2} \underline{dS} \cdot \underline{D} \quad (32)$$

is applied to the contour  $C_2$  and the surface  $S_2$ , as shown in Figure 6b, the rate of change of the current  $I$  along the cable is

$$\frac{dI(x)}{dx} = - j\omega \hat{e} E(x) (2\pi r_{\omega}) \quad (33)$$

where the current  $I$ , per unit length in the  $x$  direction, is defined by

$$I(x) = \oint \underline{dl} \cdot \underline{H} \quad (34)$$

#### 3.4.3 Transmission Line Model

The above equations developed from Faraday's Law and Amperes' Law are put into the form of the standard Telegrapher's Equation on a distributed transmission line

$$\frac{dV(x)}{dx} = - Z I(x) + E_s \quad (35)$$

$$\frac{dI(x)}{dx} = - Y V(x) \quad (36)$$

where the series impedance  $Z$  and the parallel admittance  $Y$  are defined by

$$Z = Z_L + Z_S \quad (37)$$

$$Y = \frac{1}{Z_T} \quad (38)$$

The surface impedance  $Z_S$  and the longitudinal and transverse impedances  $Z_L$  and  $Z_T$  are defined by

$$Z_S = \frac{E(x)}{I(x)} = \frac{E'(x) + E_S}{I(x)} \quad (39)$$

and

$$Z_L = j\omega \frac{\phi(x)}{I(x)} = Z_{L1} + Z_{L2} \quad (40)$$

$$Z_T = \frac{V(x)}{\frac{dI(x)}{dx}} = Z_{T1} + Z_{T2} \quad (41)$$

In (39),  $E'$  denotes the secondary induced field due to the primary return stroke field  $E_S$ . The longitudinal impedance  $Z_L$  is later decomposed into the sum of two terms  $Z_{L1}$  and  $Z_{L2}$ . Similarly, the transverse impedance  $Z_T$  is later decomposed into the sum of two terms  $Z_{T1}$  and  $Z_{T2}$ .

Strictly speaking, the impedances are functions of  $x$ . However, only a moderate variation with respect to  $x$  over the major portions of the wire is expected, except near the points where the current varies rapidly. Furthermore, it can be shown that when the fields under consideration are travelling waves, these impedances are truly independent of  $x$ . Thus, a good approximation to the impedances can be obtained by considering the natural modes guided by the coaxial configuration, as shown in Figure 7.

### 3.5 Natural Modes

To determine the exact form of the natural modes, the potentials in the earth, insulator, and outer armor of the wire are expanded as

$$\Pi_\omega = \int_{-\infty}^{+\infty} d\zeta F_\omega(\zeta) J_0(\xi_\omega r) e^{-j\zeta x}, \quad (r < r_\omega) \quad (42)$$

$$\Pi_i = \int_{-\infty}^{+\infty} d\zeta [F'_i(\zeta) J_0(\xi_i r) + F''_i(\zeta) Y_0(\xi_i r)] e^{-j\zeta x}, \quad (r_\omega < r < r_i) \quad (43)$$

$$\Pi_e = \int_{-\infty}^{+\infty} d\zeta F_e(\zeta) H_0^{(2)}(\xi_e r) e^{-j\zeta x}, \quad (r > r_i) \quad (44)$$

where  $F$ ,  $F'_i$ ,  $F''_i$ ,  $F_e$  are the unknown expansion coefficients determined

from the dispersion relationship. The transverse wave numbers  $\xi_\omega$ ,  $\xi_i$ , and  $\xi_e$  are defined by

$$\xi_\omega = \sqrt{k_\omega^2 - \zeta^2}$$

$$\xi_i = \sqrt{k_i^2 - \zeta^2}$$

$$\xi_e = \sqrt{k_e^2 - \zeta^2}$$

where the wave numbers  $k_\omega$ ,  $k_i$ , and  $k_e$  are defined by

$$k_\omega = \omega \sqrt{\mu_\omega (\epsilon_\omega + j \frac{\sigma_\omega}{\omega})}$$

$$k_i = \omega \sqrt{\mu_i (\epsilon_i + j \frac{\sigma_i}{\omega})}$$

$$k_e = \omega \sqrt{\mu_e (\epsilon_e + j \frac{\sigma_e}{\omega})}$$

The impedances  $Z_S$ ,  $Z_{L1}$ ,  $Z_{L2}$ ,  $Z_{T1}$ , and  $Z_{T2}$  defined in (39)-(41) may be expressed in terms of the unknown expansion coefficients  $F_\omega$ ,  $F'_i$ ,  $F''_i$ , and  $F_e$ :

$$Z_S(x) = \frac{\int F_\omega(\zeta) \xi_c^2 J_0(\xi_c b) e^{-j\zeta x} d\zeta}{2\pi b (\sigma_c + j\omega\epsilon_c) \int F_\omega(\zeta) \xi_c J_1(\xi_c b) e^{-j\zeta x} d\zeta} \quad (45)$$

and

$$Z_{L1}(x) = \frac{-j\omega\mu_i (\sigma_i + j\omega\epsilon_i) \int \{F'_i(\zeta) [J_0(\xi_1 c) - J_0(\xi_1 b)] + F''_i(\zeta) [Y_0(\xi_1 c) - Y_0(\xi_1 b)]\} e^{-j\zeta x} d\zeta}{2\pi b (\sigma_c + j\omega\epsilon_c) \int F_\omega(\zeta) \xi_c J_1(\xi_c b) e^{-j\zeta x} d\zeta} \quad (46)$$

$$Z_{L2}(x) = \frac{j\omega\mu_e (\sigma_e + j\omega\epsilon_e)}{2\pi b (\sigma_c + j\omega\epsilon_c)} \frac{\int F_e(\zeta) H_0^{(2)}(\xi_e c) e^{-j\zeta x} d\zeta}{\int F_\omega(\zeta) \xi_c J_1(\xi_c b) e^{-j\zeta x} d\zeta} \quad (47)$$

and

$$Z_{T1}(x) = \frac{-\int \{F'_i(\zeta) [J_0(\xi_1 c) - J_0(\xi_1 b)] + F''_i(\zeta) [Y_0(\xi_1 c) - Y_0(\xi_1 b)]\} \zeta e^{-j\zeta x} d\zeta}{2\pi b (\sigma_c + j\omega\epsilon_c) \int \zeta \xi_c F_\omega(\zeta) J_1(\xi_c b) e^{-j\zeta x} d\zeta} \quad (48)$$

$$Z_{T2}(x) = \frac{\int \zeta F_e(\zeta) H_0^{(2)}(\xi_e c) e^{-j\zeta x} d\zeta}{2\pi b (\sigma_c + j\omega\epsilon_c) \int \zeta \xi_c F_\omega(\zeta) J_1(\xi_c b) e^{-j\zeta x} d\zeta}$$

The dispersion relationship for the coaxial configuration, as determined from the boundary conditions on the fields at the wire-insulator interface and the insulator-earth interface, are

$$\begin{bmatrix} (\sigma_c + j\omega\epsilon_c)\xi_c J_1(\xi_c b) & -(\sigma_i + j\omega\epsilon_i)\xi_i J_1(\xi_i b) & -(\sigma_i + j\omega\epsilon_i)\xi_i Y_i(\xi_i b) & 0 \\ 0 & (\sigma_i + j\omega\epsilon_i)\xi_i J_1(\xi_i c) & (\sigma_i + j\omega\epsilon_i)\xi_i Y_i(\xi_i c) & -(\sigma_e + j\omega\epsilon_e)\xi_e H_1^{(2)}(\xi_e c) \\ \xi_c^2 J_0(\xi_c b) & -\xi_i^2 J_0(\xi_i b) & -\xi_i^2 Y_0(\xi_i b) & 0 \\ 0 & \xi_i^2 J_0(\xi_i c) & \xi_i^2 Y_0(\xi_i c) & -\xi_e^2 H_0^{(2)}(\xi_e c) \end{bmatrix} \begin{bmatrix} F_\omega \\ F_i' \\ F_i'' \\ F_e \end{bmatrix} = 0$$

which is a homogeneous matrix equation for the unknown expansion coefficients. For a non-trivial solution for the expansion coefficients, the determinant of the above matrix must be zero, i.e.

$$\Delta_1 = \Delta_2$$

where

$$\Delta_1 = \frac{\frac{\sigma_c + j\omega\epsilon_c}{\sigma_1 + j\omega\epsilon_1} \frac{\xi_1}{\xi_c} J_1(\xi_c b) J_0(\xi_1 b) - J_0(\xi_c b) J_1(\xi_1 b)}{\frac{\sigma_c + j\omega\epsilon_c}{\sigma_1 + j\omega\epsilon_1} \frac{\xi_1}{\xi_c} J_1(\xi_c b) Y_0(\xi_1 b) - J_0(\xi_c b) Y_1(\xi_1 b)} \quad (51)$$

$$\Delta_2 = \frac{\frac{\sigma_e + j\omega\epsilon_e}{\sigma_i + j\omega\epsilon_i} \frac{\xi_1}{\xi_e} H_1^{(2)}(\xi_e c) J_0(\xi_1 c) - H_0^{(2)}(\xi_e c) J_1(\xi_1 c)}{\frac{\sigma_e + j\omega\epsilon_e}{\sigma_i + j\omega\epsilon_i} \frac{\xi_1}{\xi_e} H_1^{(2)}(\xi_e c) Y_0(\xi_1 c) - H_0^{(2)}(\xi_e c) Y_1(\xi_1 c)} \quad (52)$$

The propagation constant  $\zeta$  determined from the above expressions is denoted by  $\zeta_0$ , which is the eigenvalue of the determinant. Also, the dependence of the expansion coefficients  $F_i'$ ,  $F_i''$ , and  $F_e$  on  $F_\omega$  is attained, i.e. the eigenfunctions of the determinant are

$$\frac{F_i'(\zeta)}{F_\omega(\zeta)} = \frac{\sigma_c + j\omega\epsilon_c}{\sigma_1 + j\omega\epsilon_1} \frac{\xi_c}{\xi_1} \frac{J_1(\xi_c b)}{J_1(\xi_1 b) - \Gamma Y_1(\xi_1 b)} \quad (53)$$

$$\frac{F_i''(\zeta)}{F_\omega(\zeta)} = \frac{-(\sigma_c + j\omega\epsilon_c)}{\sigma_1 + j\omega\epsilon_1} \frac{\xi_c}{\xi_1} \frac{J_1(\xi_c b)}{J_1(\xi_1 b) - \Gamma Y_1(\xi_1 b)} \quad (54)$$

$$\frac{F_e(\zeta)}{F_\omega(\zeta)} = \frac{\sigma_c + j\omega\epsilon_c}{\sigma_1 + j\omega\epsilon_1} \frac{\xi_1 \xi_c}{\xi_e^2} \frac{J_1(\xi_c b) [J_0(\xi_1 c) - \Gamma Y_0(\xi_1 c)]}{H_0^{(2)}(\xi_e c) [J_1(\xi_1 b) - \Gamma Y_1(\xi_1 b)]} \quad (55)$$

where

$$\Gamma = \frac{\frac{\sigma_c + j\omega\epsilon_c}{\sigma_1 + j\omega\epsilon_1} \frac{\xi_1}{\xi_c} J_1(\xi_c b) J_0(\xi_1 b) - J_0(\xi_c b) J_1(\xi_1 b)}{\frac{\sigma_c + j\omega\epsilon_c}{\sigma_1 + j\omega\epsilon_1} \frac{\xi_1}{\xi_c} J_1(\xi_c b) Y_0(\xi_1 b) - J_0(\xi_c b) Y_1(\xi_1 b)} \quad (56)$$

When the current on the wire is a traveling wave of the form

$$I(x) = I_0 e^{-j\zeta_0 x} = \int_{-\infty}^{+\infty} d\zeta I_0 \delta(\zeta - \zeta_0) e^{-j\zeta x} \quad (57)$$

where  $I_0$  is a constant, the remaining eigenfunction  $F_\omega$  is determined to be

$$F_\omega(\zeta) = \frac{I_0 \delta(\zeta - \zeta_0)}{\sqrt{2\pi} b(\sigma_c + j\omega\epsilon_c) \epsilon_c J_1(\epsilon_c b)} \quad (58)$$

Therefore, due to the presence of the delta function in  $F_\omega$ , the integrals in (45)-(49) for the impedances are easily evaluated to give

$$Z_s = \left. \frac{\epsilon_c J_0(\epsilon_c b)}{2\pi b(\sigma_c + j\omega\epsilon_c) J_1(\epsilon_c b)} \right|_{\zeta=\zeta_0} \quad (59)$$

and

$$Z_{L1} = \left. \frac{-j\omega\mu_1(\sigma_1 + j\omega\epsilon_1)}{2\pi b(\sigma_c + j\omega\epsilon_c)} \frac{F'_1(\zeta)[J_0(\epsilon_1 c) - J_0(\epsilon_1 b)] + F''_1(\zeta)[Y_0(\epsilon_1 c) - Y_0(\epsilon_1 b)]}{\epsilon_c F_\omega(\zeta) J_1(\epsilon_c b)} \right|_{\zeta=\zeta_0} \quad (60)$$

$$Z_{L2} = \left. \frac{j\omega\mu_e(\sigma_e + j\omega\epsilon_e)}{2\pi b(\sigma_c + j\omega\epsilon_c)} \frac{F_e(\zeta)}{F_\omega(\zeta)} \frac{H_0^{(2)}(\epsilon_e c)}{\epsilon_c J_1(\epsilon_c b)} \right|_{\zeta=\zeta_0} \quad (61)$$

and

$$Z_{T1} = \left. \frac{-1}{2\pi b(\sigma_c + j\omega\epsilon_c)} \frac{F'_1(\zeta)[J_0(\epsilon_1 c) - J_0(\epsilon_1 b)] + F''_1(\zeta)[Y_0(\epsilon_1 c) - Y_0(\epsilon_1 b)]}{\epsilon_c F_\omega(\zeta) J_1(\epsilon_c b)} \right|_{\zeta=\zeta_0} \quad (62)$$

$$Z_{T2} = \left. \frac{1}{2\pi b(\sigma_c + j\omega\epsilon_c)} \frac{F_e(\zeta)}{F_\omega(\zeta)} \frac{H_0^{(2)}(\epsilon_e c)}{\sigma_c J_1(\epsilon_c b)} \right|_{\zeta=\zeta_0} \quad (63)$$

### 3.6 Green's Functions

The voltage and current waves on the outermost conductor of a coaxial transmission line excited by a distributed voltage source due to a lightning discharge are now summarized.

The voltage  $V(x)$  and the current  $I(x)$  on the transmission line due to a distributed voltage source  $V(x')$  are determined by the superposition integrals

$$V(x) = \int_0^{\ell} dx' G_V(x; x') V(x') \quad (64)$$

$$I(x) = \int_0^{\ell} dx' G_I(x; x') V(x') \quad (65)$$

where  $V(x')$  is the field due to the lightning stroke, i.e.

$$V(x') = E_S \quad (66)$$

and the voltage Green's function  $G_V(x, x')$  and the current Green's function  $G_I(x, x')$  are known to be

$$G_V(x, x') = \begin{cases} V^-(e^{+\gamma x} + \Gamma^- e^{-\gamma x}) & (x < x') \\ V^+(e^{-\gamma x} + \Gamma^+ e^{+\gamma x} e^{-2\gamma \ell}) & (x > x') \end{cases} \quad (67)$$

$$(68)$$

$$G_I(x, x') = \begin{cases} -\frac{V^-}{Z_c} (e^{+\gamma x} - \Gamma^- e^{-\gamma x}) & (x < x') \\ +\frac{V^+}{Z_c} (e^{-\gamma x} - \Gamma^+ e^{+\gamma x} e^{-2\gamma \ell}) & (x > x') \end{cases} \quad (69)$$

$$(70)$$

The unknown constants  $V^\pm$ , which are the magnitudes of the forward and backward traveling waves, are determined from the boundary conditions imposed by the continuity of the current wave and the discontinuity of the voltage wave at  $x = x'$ , i.e.

$$V^- = + \frac{e^{-\gamma x'} - \Gamma^+ e^{+\gamma x'} e^{-2\gamma \ell}}{\Delta} \quad (71)$$

$$V^+ = - \frac{e^{+\gamma x'} - \Gamma^- e^{-\gamma x'}}{\Delta} \quad (72)$$

where the reflection coefficients  $\Gamma^\pm$  are defined by

$$\Gamma^- = \frac{Z^- - Z_c}{Z^- + Z_c} \quad (73)$$

$$\Gamma^+ = \frac{Z^+ - Z_c}{Z^+ + Z_c} \quad (74)$$

and the determinant  $\Delta$  is defined by

$$\Delta = -2(1 - \Gamma^- \Gamma^+ e^{-2\gamma \ell}) \quad (75)$$

The propagation constant  $\gamma$  and the characteristic impedance  $Z_c$  are defined by

$$\gamma = \sqrt{YZ} \quad (76)$$

$$Z_c = \sqrt{\frac{Z}{Y}} \quad (77)$$

where the series impedance  $Z$  and the shunt admittance  $Y$  were defined earlier in (37) and (38).

### 3.7 Transfer Functions

Once the induced current and voltage surges on the outer conductor ( $\rho = \rho_>$ ) of a coaxial conductor are known due to a nearby lightning discharge, the induced current and voltage surges on the inner conductor ( $\rho = \rho_<$ ) are determined via the use of the impedance transfer functions for a coaxial cable, i.e.

$$E_x \Big|_{\rho=\rho_>} = Z_{ee} I_{\text{ext}} + Z_{ei} I_{\text{int}} \quad (78)$$

$$E_x \Big|_{\rho=\rho_<} = Z_{ie} I_{\text{ext}} + Z_{ii} I_{\text{int}} \quad (79)$$

where the internal and external impedances are defined by

$$Z_{ee} = \frac{k}{-j\omega\epsilon} \frac{1}{2\pi\rho_>} \frac{\Delta_{ee}}{L} \quad (80)$$

$$Z_{ei} = \frac{1}{-j\pi\epsilon} \frac{1}{2\pi} \frac{1}{\rho_<\rho_>} \frac{1}{L} \quad (81)$$

$$Z_{ie} = \frac{1}{-j\omega\epsilon} \frac{1}{2\pi} \frac{1}{\rho_<\rho_>} \frac{1}{L'} \quad (82)$$

$$Z_{ii} = \frac{k}{-j\omega\epsilon} \frac{1}{2\pi\rho_<} \frac{\Delta_{ii}}{L'} \quad (83)$$

and

$$\Delta_{ee} = Y_1(k\rho_<)J_0(k\rho_<) + J_1(k\rho_<)Y_0(k\rho_>) \quad (84)$$

$$\Delta_{ii} = Y_1(k\rho_>)J_0(k\rho_>) + J_1(k\rho_>)Y_0(k\rho_<) \quad (85)$$

and

$$\Delta = J_1(k\rho_>)Y_1(k\rho_<) - J_1(k\rho_<)Y_1(k\rho_>) \quad (86)$$

$$L' = J_1(k\rho_<)Y_1(k\rho_>) - J_1(k\rho_>)Y_1(k\rho_<) \quad (87)$$

Since the coaxial cable may be composed of several concentric shells rather than just one, as depicted above, the above formulas may be used recursively to obtain the current and voltages on the innermost cylinder due to the current and voltages on the outermost cylinder.

## CHAPTER 4

### The Computer Program

#### 4.1 Introduction

The "LPS" Program has been written in FORTRAN IV language for the CDC Cyber 74 computer to produce solutions to the electromagnetic coupling problems which are outlined above.

Several factors were given consideration in the design of the program. Efficient coding was used to reduce computer run time as much as possible. Full single word accuracy and, where necessary, double precision accuracy was utilized in the calculation of special functions. Single precision accuracy on the Cyber maintains fifteen digits of accuracy. Maximum use was made of core storage. The sizes of the basic solution arrays (up to 100 x 100 complex) were determined so that secondary storage devices such as tapes or disk packs do not necessarily have to be utilized in running the program.

#### 4.2 Glossary of the Routines

The program operates via a MAIN program and numerous auxiliary routines, e.g., a runstream, program sequences, subroutines, functions, and a library, which are listed in a flow chart contained in Figure 8. Standard I/O and mathematical routines, e.g., intrinsic functions, are assumed to be available through the FORTRAN operating system.

The MAIN program controls the overall runstream and sets up the parameter, type, and dimension statements and the common blocks.

The runstream calls the program sequences, which in turn call various subroutines and functions. A library of special functions and operations was developed to support the program.

#### 4.3 The Input Routine

Subroutine PS10 reads the user's control information, prints out headings, and obtains information for all operations. The input cards and their formats are listed in Table 1.

## CHAPTER 5

### Numerical Results

Several average lightning channel conditions and representative buried cable geometries are examined. The results are conveniently displayed via numerous graphs of the time histories (time-domain results) and the frequency spectra (frequency-domain results) of the resulting transient voltage and current pulses.

In particular, a typical RG-58/U coaxial cable buried one meter below the ground in wet soil is considered. The cable is assumed to be shielded and armored and is coated with a protective layer of insulation. The radius of the wire is 0.813 mm, the inner and outer radii of the shields are, respectively, 2.953 mm and 3.953 mm, the inner and outer radii of the armor are, respectively, 4.953 mm and 5.953 mm, and the outer radius of the sheath is 6.953 mm. The relative permeabilities of all of the insulators and conductors are assumed to be unity; whereas, the relative permittivities of the stabilized polyethylene spacers are assumed to be 2.26. The absolute conductivities of the copper conductors are assumed to be  $5.7 \times 10^7$  mhos/m; the absolute conductivities of the stabilized polyethylene spacers are assumed to be  $1.0 \times 10^{-8}$  mhos/m. The earth is assumed to be wet with relative permeability 1.0, relative permittivity 5.0, and absolute conductivity  $1.5 \times 10^{-3}$  mhos/m. A "double exponential" lightning channel pulse is assumed, with typical rise-times and half-life decay times on the order of 1.2  $\mu$ s and 50  $\mu$ s, respectively. A peak current of 20 KA is also assumed. The lightning channel is considered to be vertical and, for comparison purposes, is assumed to terminate at  $d = 0$  m, 10m, 50m, 100m, 200m, 500m, 1km, 2km, 5km, and 10km from the cable, at a point near the end of the cable. The cable is assumed to be 1 km in length.

Particular attention is paid to the magnitudes of the voltage and current waveforms at both ends of the line and at the middle of the line, i.e., at  $x = 0$  m, 500 m, and 1 km.

As an example, Figures 9 through 16 contain plots of the computer output which resulted from the input data contained in Appendix B. In particular, Figure 9 shows the frequency-domain distribution of the external excitation, i.e., axial electric field on the outer conductor of the cable, as a function of position at various frequencies due to the lightning stroke directly over the load. Figure 10 shows the frequency-domain distribution of the induced current transient on the outer conductor of the cable as a function of position at various frequencies due to the distributed external excitation of the lightning stroke. Figure 11 shows the frequency spectrum of the transfer ratio for the current or voltage between the inner and outer conductors of the cable. Figure 12 shows the frequency-domain distribution of the internal excitation, i.e., the axial electric field on the inner conductor of the cable, as a function of position at various frequencies due to the imperfect shielding of the armor and the shield. Figures 13 and 14 show, respectively, the frequency spectra of the current and voltage pulses on the inner conductor of the cable; and, Figures 15 and 16 show, respectively, the time histories of the current and voltage pulses on the inner conductor of the cable.

Table 2 contains the values of the peak current and voltage and the rise times and the half-life decay times of the pulses on the inner conductor of the cable at the loads, and at the midpoint of the cable as a function of the distance of the lightning stroke from the cable with the other parameters of the model held constant. The lightning stroke occurs near the end of the cable. The program is capable of varying all of the parameters in the model simultaneously; but, only this one case is studied here. This case is of interest and is

treated in particular since it is very similar to the conditions which actually exist during a thunderstorm, i.e., the distance from the cable to the lightning stroke varies from stroke to stroke; whereas, the remaining parameters of the model are fixed at the initial time of installation of the system.

An examination of Table 2 reveals that the ability of the lightning stroke to induce current and voltage transients on the inner conductor of the cable at the load decreases slowly with increasing distance between the lightning stroke and the cable, as one would naturally expect. We also note that there is relatively little difference between the values corresponding to lightning strikes occurring near the ends and near the middle of the cable, which indicates that the coupling mechanism is relatively insensitive to axial displacements along the length of the cable.

The maximum current and voltage peaks observed on the inner conductor of the cable at the load are 652 V and 13.1 A, and these maxima both occur when the stroke is directly over the cable.

## CHAPTER 6

### Conclusions

This report presents the numerical results of a lightning protection study which was performed for the Federal Aviation Administration (FAA).

In particular, this study considers the lightning induced transients on a buried shielded and armored coaxial transmission line.

A lightning stroke to ground (not an intra or inter cloud stroke) near a buried coaxial cable can induce current and voltage surges on the inner core of the cable, even if the cable has an outer armor and an inner shield.

The external coupling mechanism, i.e., from the terminal ground point of the "vertical" lightning channel to the outer conductor of the cable, is either arcing, i.e., by dielectric breakdown through the earth to the cable jacket, for a "direct" strike, or conductive energization of the cable jacket through the conductive earth, for an "indirect" strike. The internal coupling mechanism, i.e., from the outer jacket to the inner core of the cable, is conductive transfer through the various metal skins and dielectric spacers, i.e., skin effects, due to the finite conductivities of these materials.

In particular, these lightning induced surges tend to cause excessive voltages to appear at the ends of the wires; and, therefore, tend to cause excessive currents to flow into the terminating equipment loads. Obviously, then, circuit disturbances or failures are likely to occur unless adequate protective measures are provided.

Since there is a tendency to move from "high voltage" electron tubes or discrete transistor circuitry to "low voltage" solid state integrated electronics, the surge protection of carbon blocks, neon bulbs, etc., (devices which are presently in service) may no longer be adequate.

#### REFERENCES

- (1) Cosel, R. M., Revay, A. W., Jr., and Stahmann, J., "FAA Lightning Protection Study: Lightning Simulation Test of Communication Cables," Report No. FAA-RD-74-41, January 1974.
- (2) Chen, C. L., "FAA Lightning Protection Study: Lightning Protection Devices," Report No. FAA-RD-74-104, April 1974.
- (3) Huddleston, G. K., Nordgard, J. D., and Larson, R. W., "FAA Lightning Protection Study: Lightning Protection Requirements for AN/GRN-27(V) Instrument Landing System," Report No. FAA-RD-74-131, April 1974.
- (4) Nordgard, J. D. and Chen, C. L., "FAA Lightning Protection Study: Lightning-induced Transients on Buried Shielded Transmission Lines," Report No. FAA-RD-75-108, June 1975.

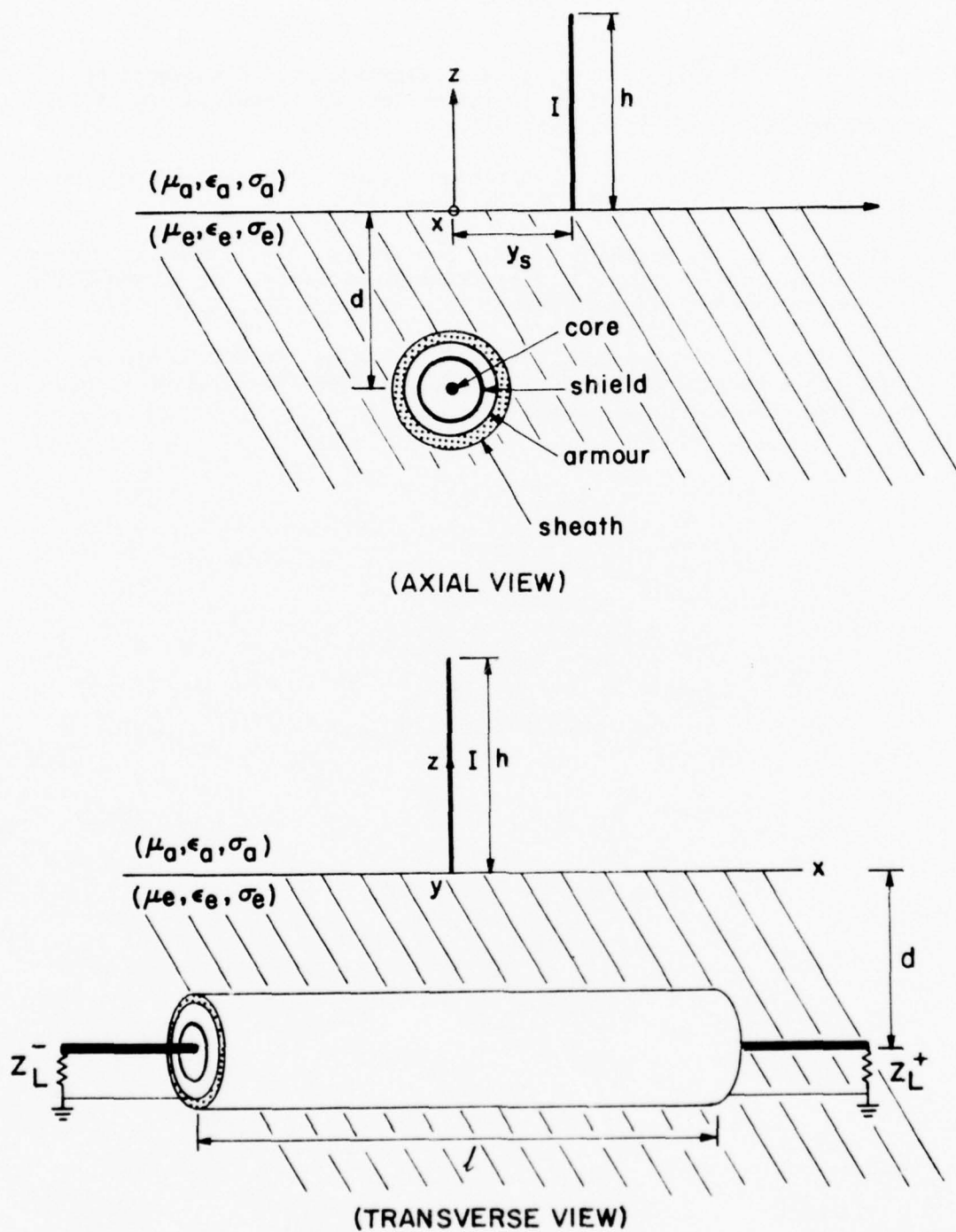


FIG. 1. PROBLEM GEOMETRY

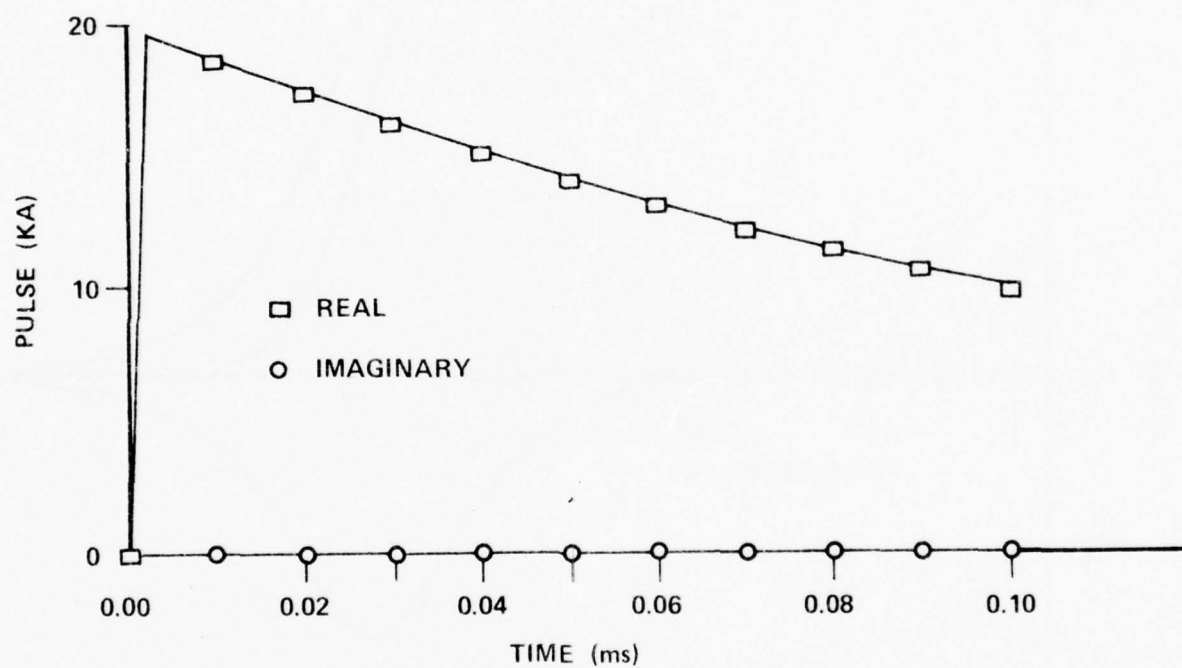


FIGURE 2. LIGHTNING PULSE (TIME DOMAIN) VS. TIME

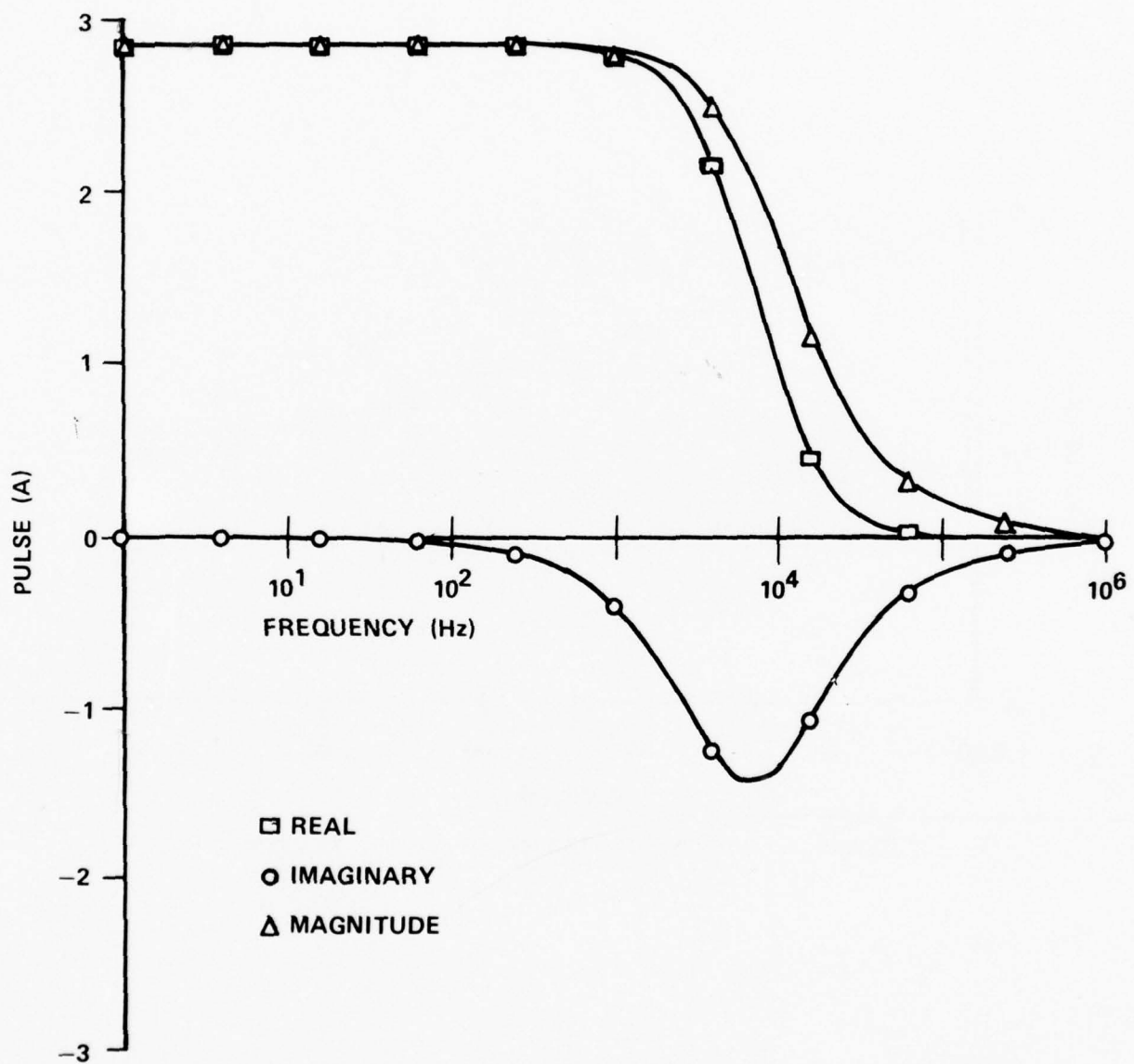


FIGURE 3. LIGHTNING PULSE (FREQUENCY DOMAIN) VS. FREQUENCY

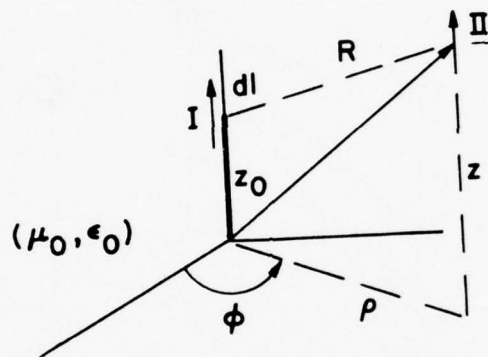


FIG. 4. A POINT DIPOLE IN FREE SPACE

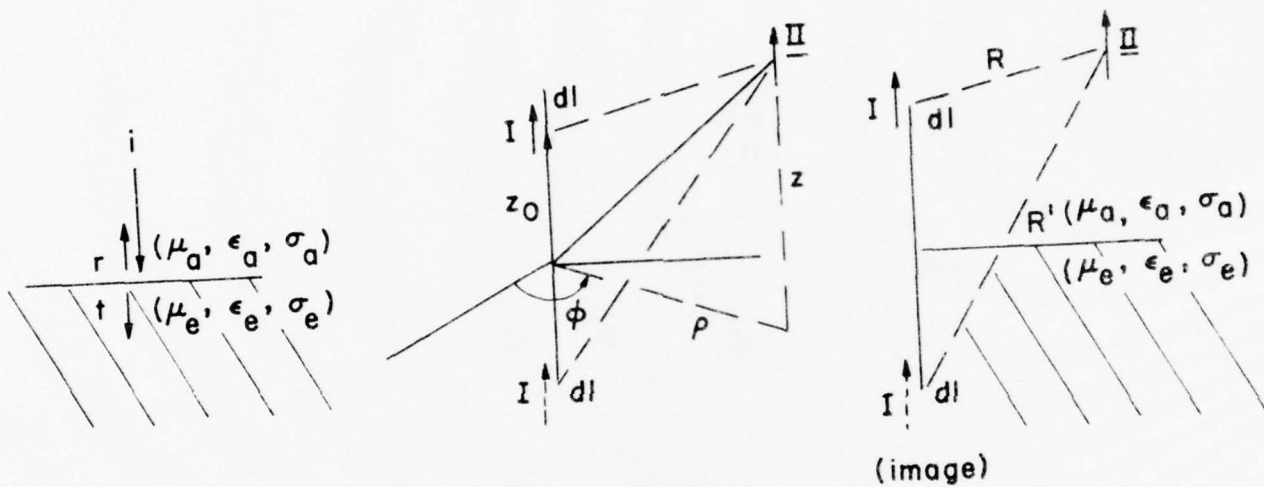


FIG. 5. A POINT DIPOLE ABOVE FLAT EARTH

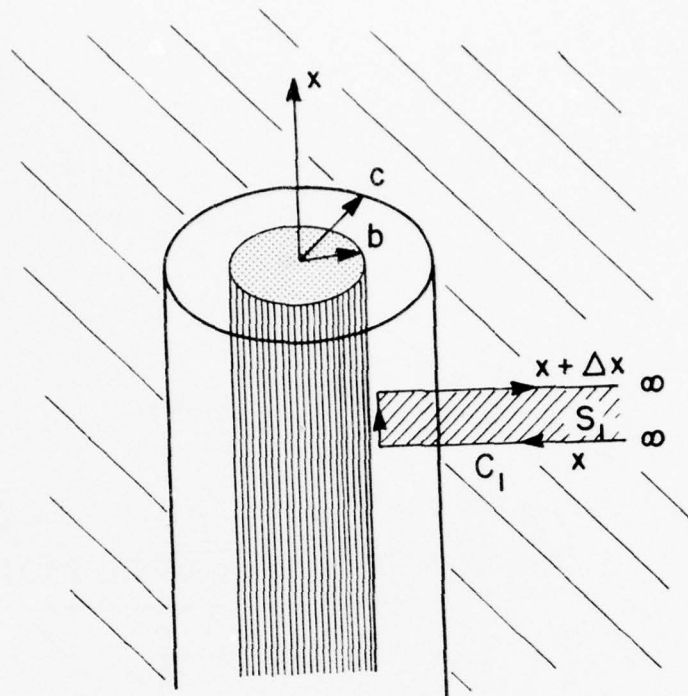


FIG. 6a. CONTOUR  $C_1$  AND SURFACE  $S_1$  IN FARADAY'S LAW

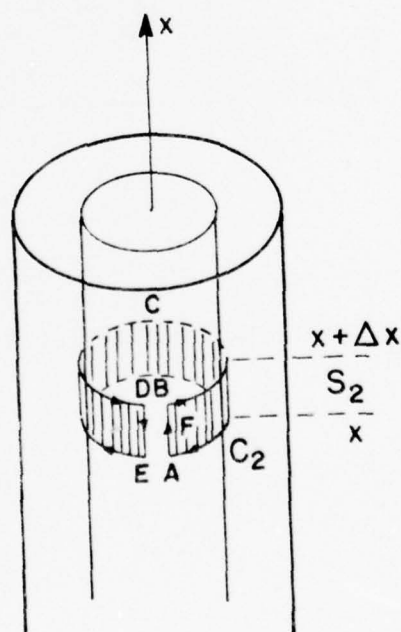


FIG. 6b. CONTOUR  $C_2$  AND SURFACE  $S_2$  IN AMPERE'S LAW

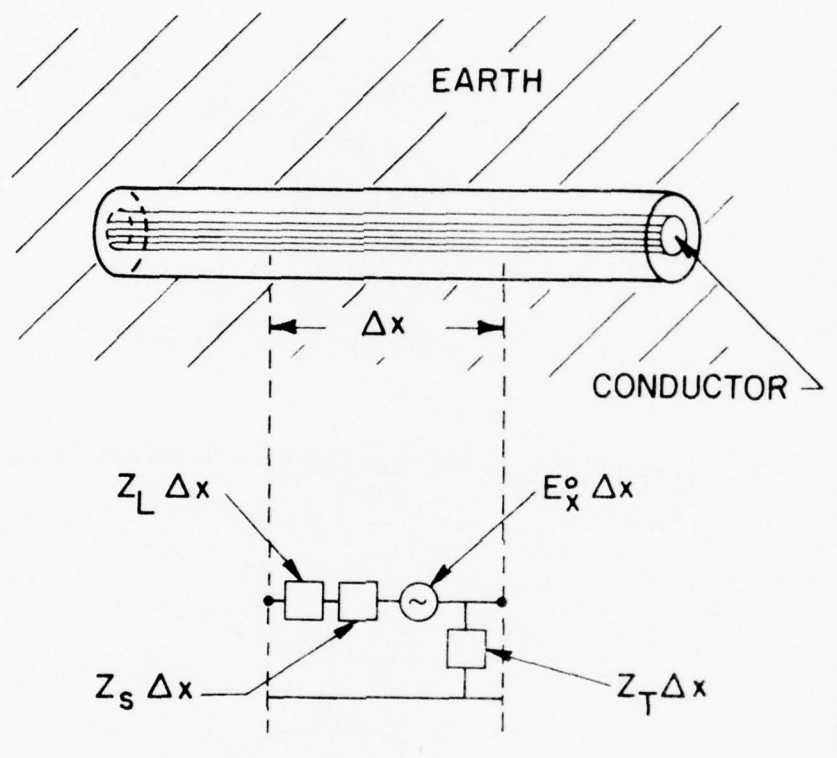
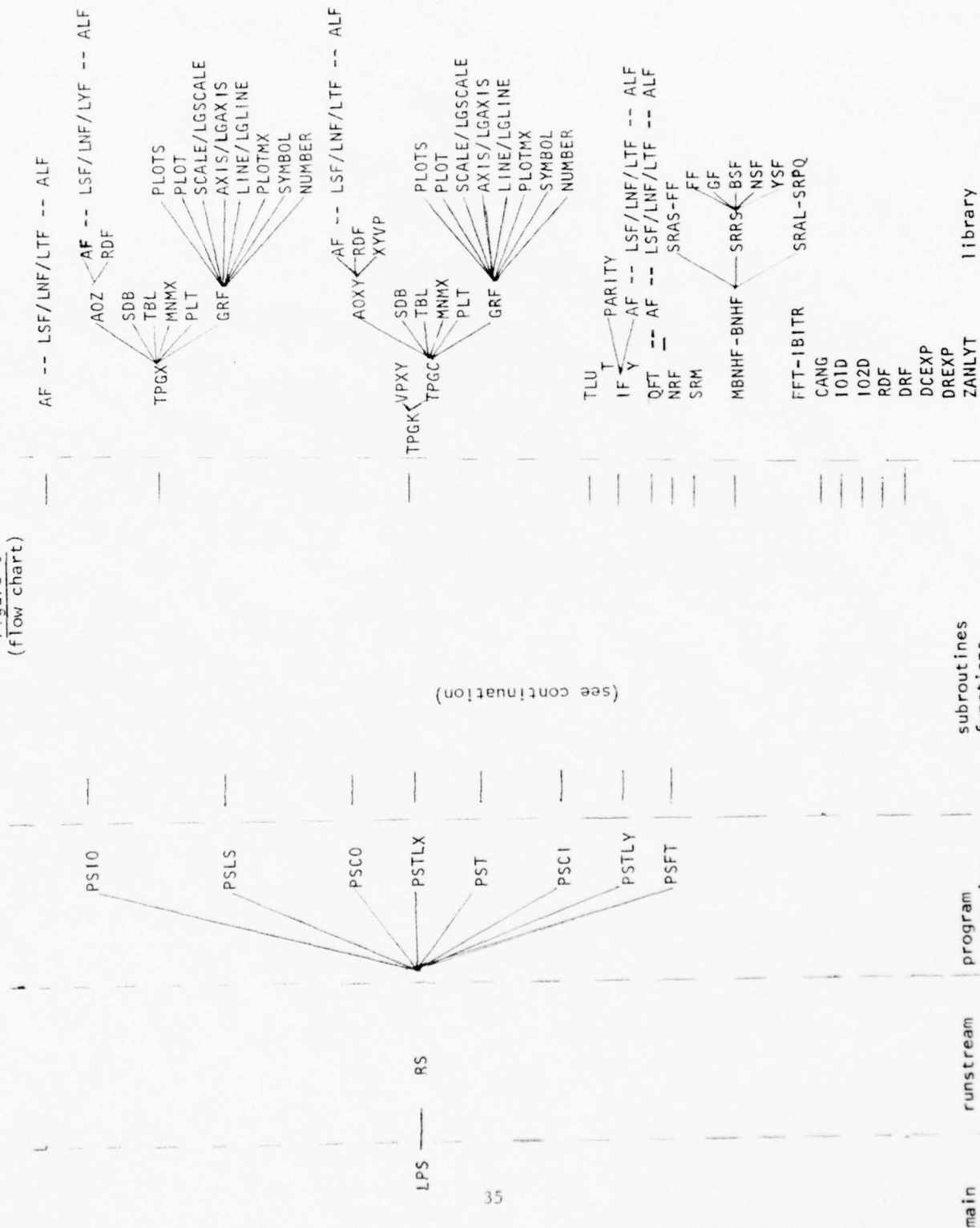


FIG. 7. DISTRIBUTED TRANSMISSION LINE MODEL

Figure 8  
(flow chart)



(see continuation)

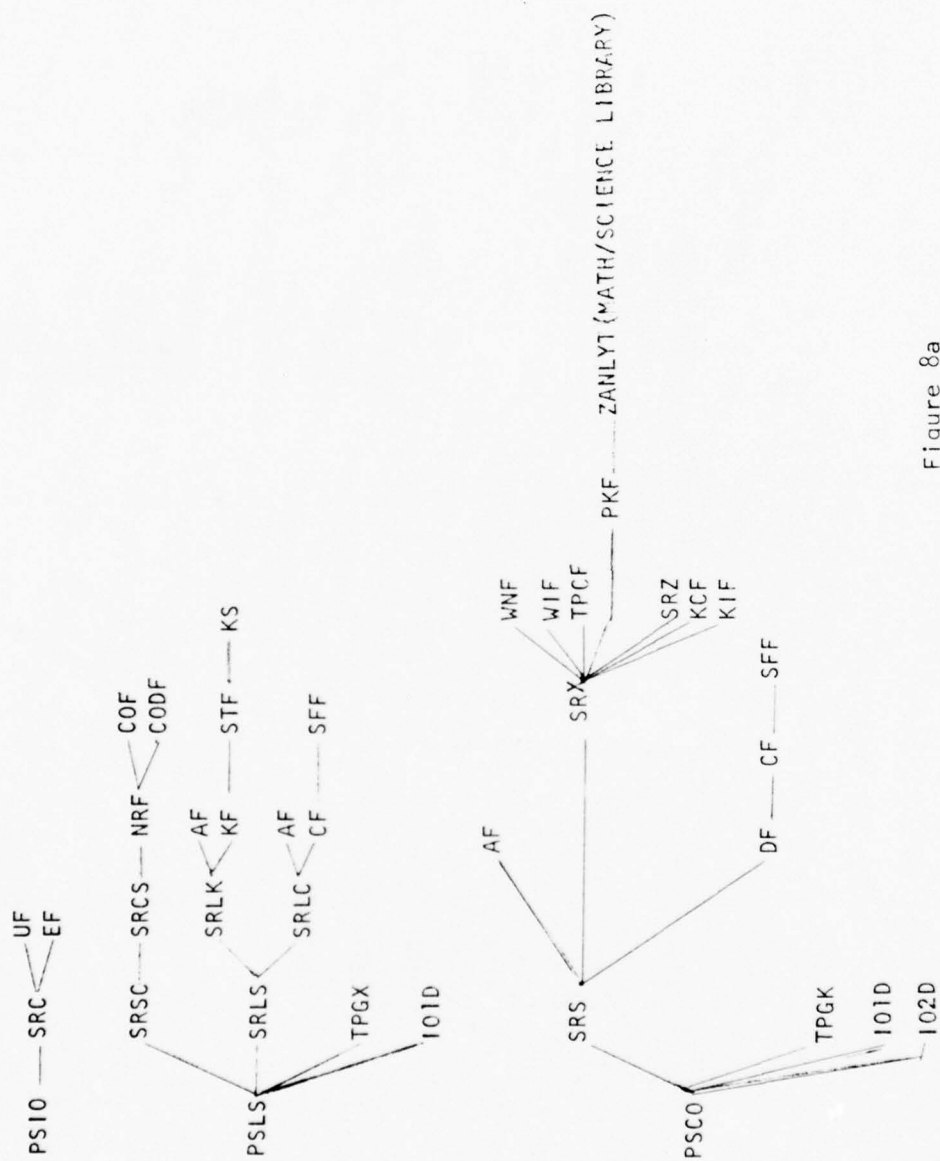


Figure 8a  
(flow chart cont.)



Figure 8b  
(flow chart cont.)

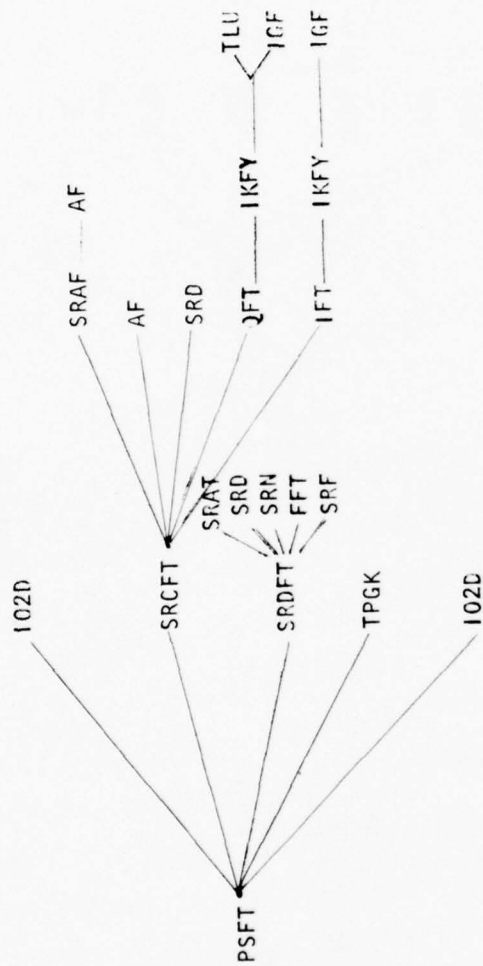


Figure 8c  
(flow chart cont.)

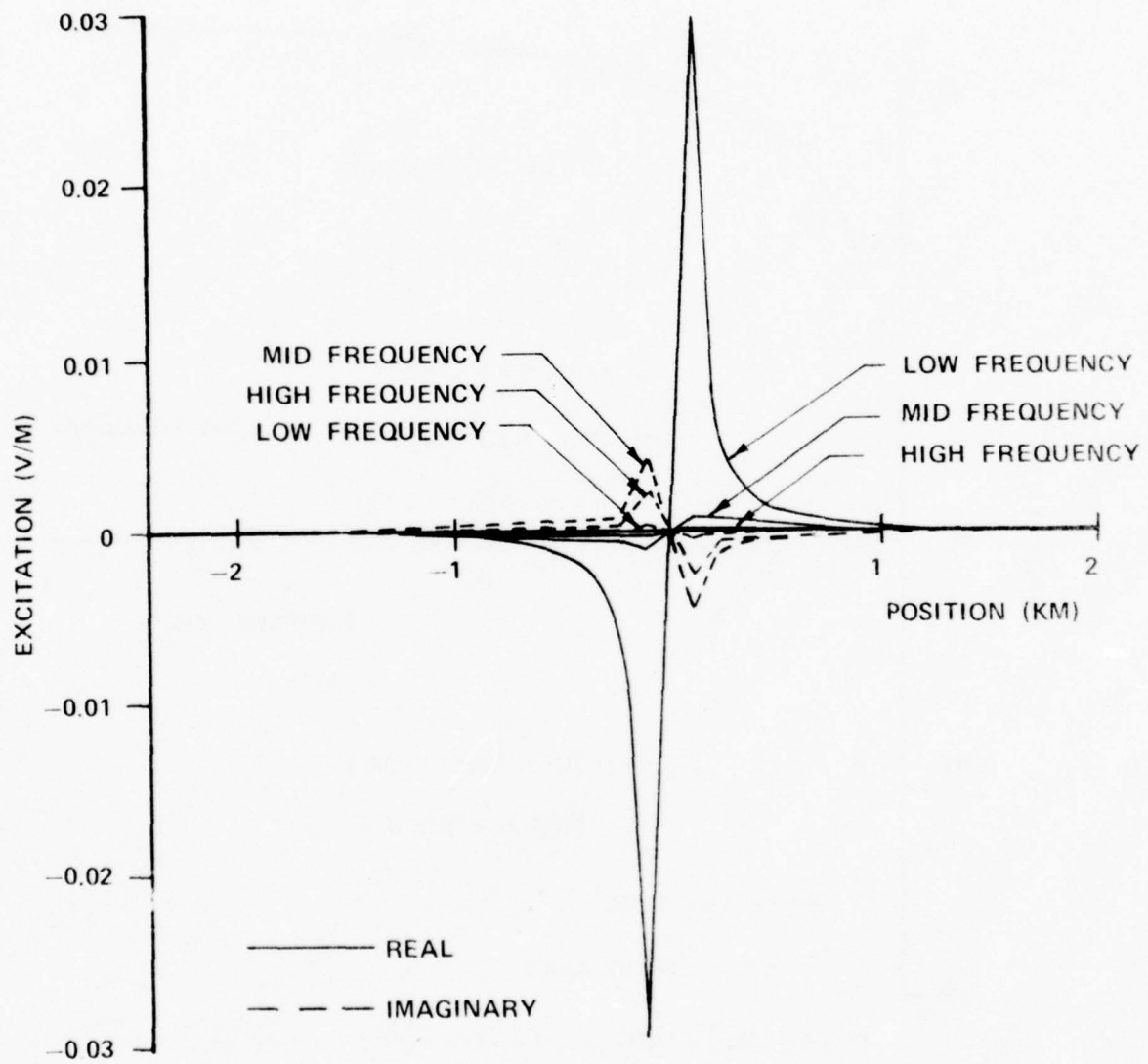


FIGURE 9. OUTER EXCITATION (FREQUENCY DOMAIN) VS. POSITION

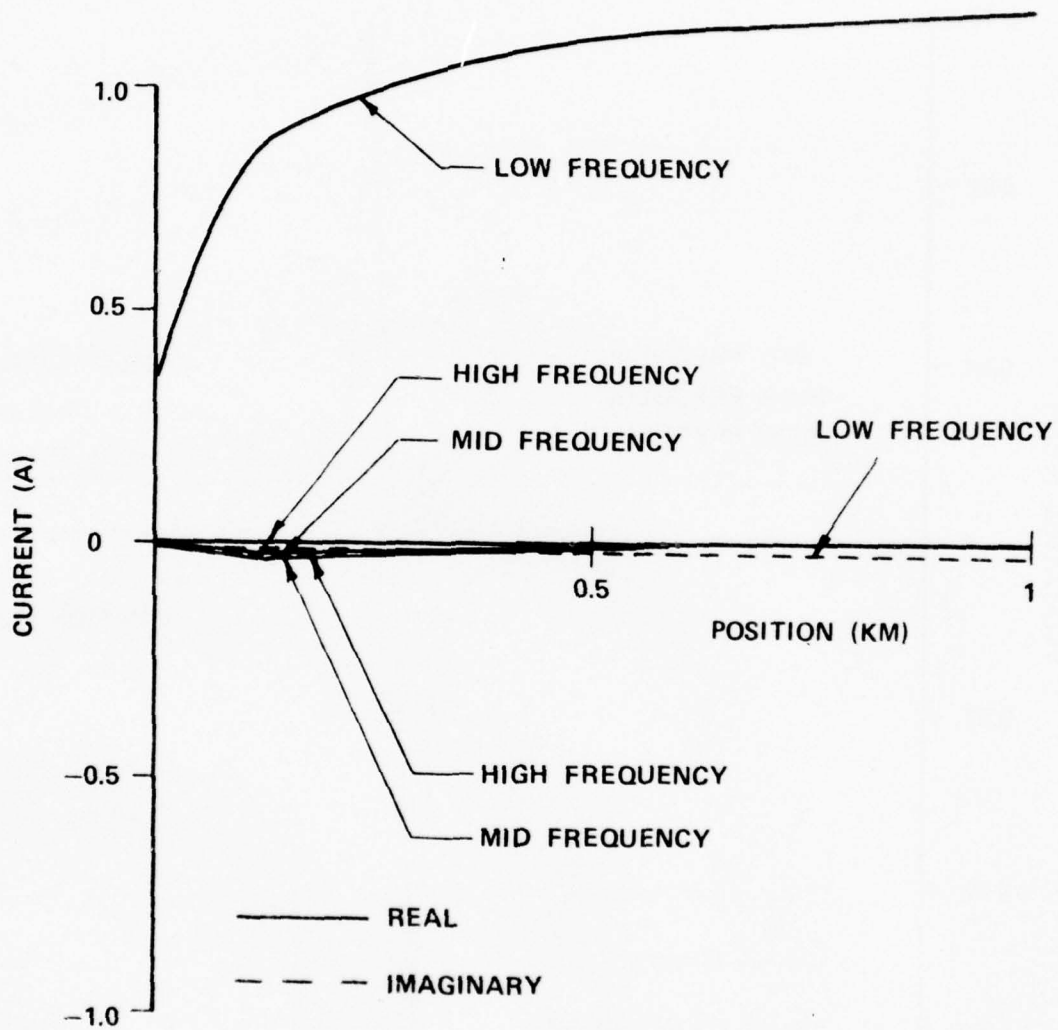


FIGURE 10. OUTER CURRENT (FREQUENCY DOMAIN) VS. POSITION

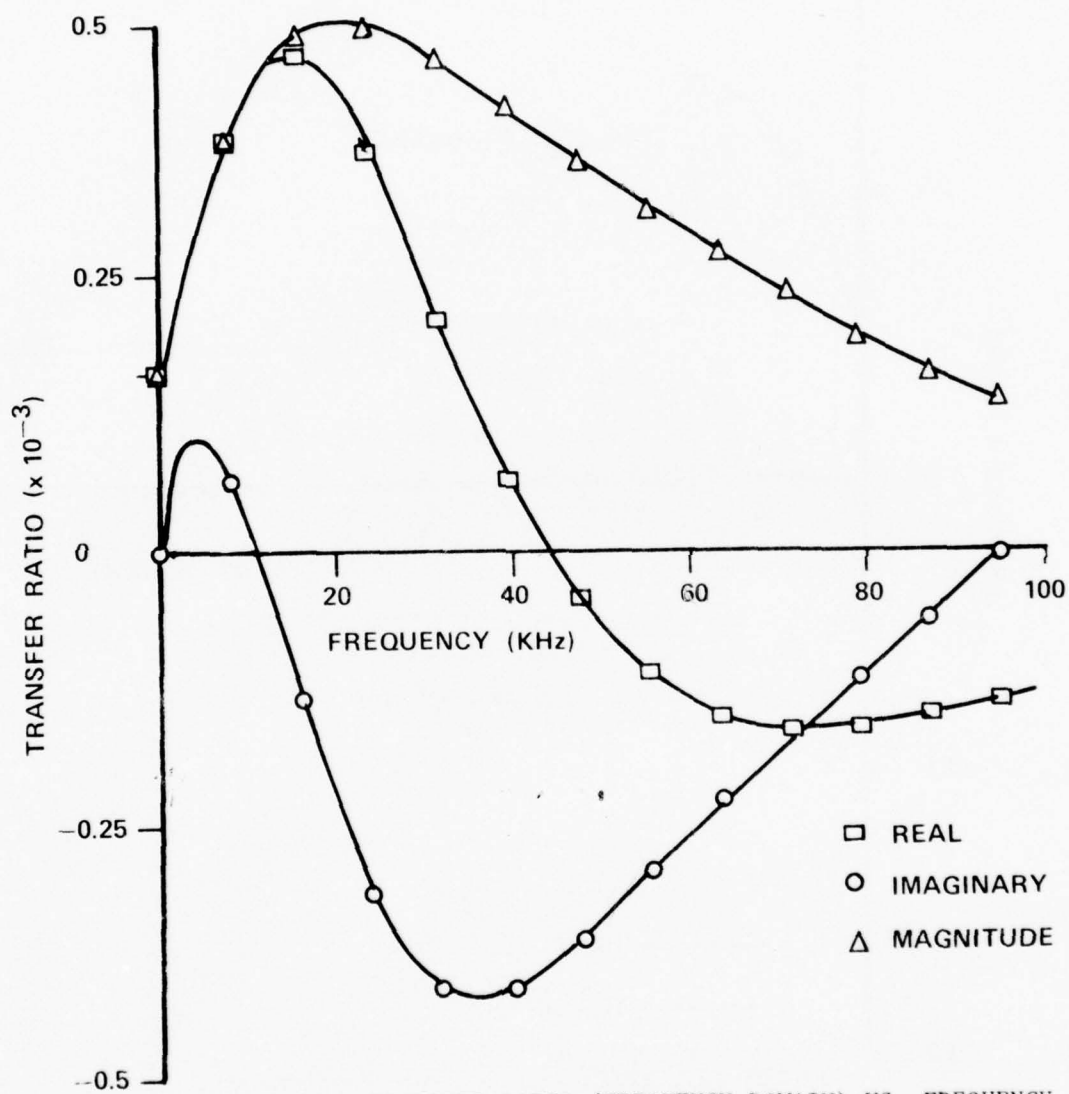


FIGURE 11. TRANSFER RATIO (FREQUENCY DOMAIN) VS. FREQUENCY

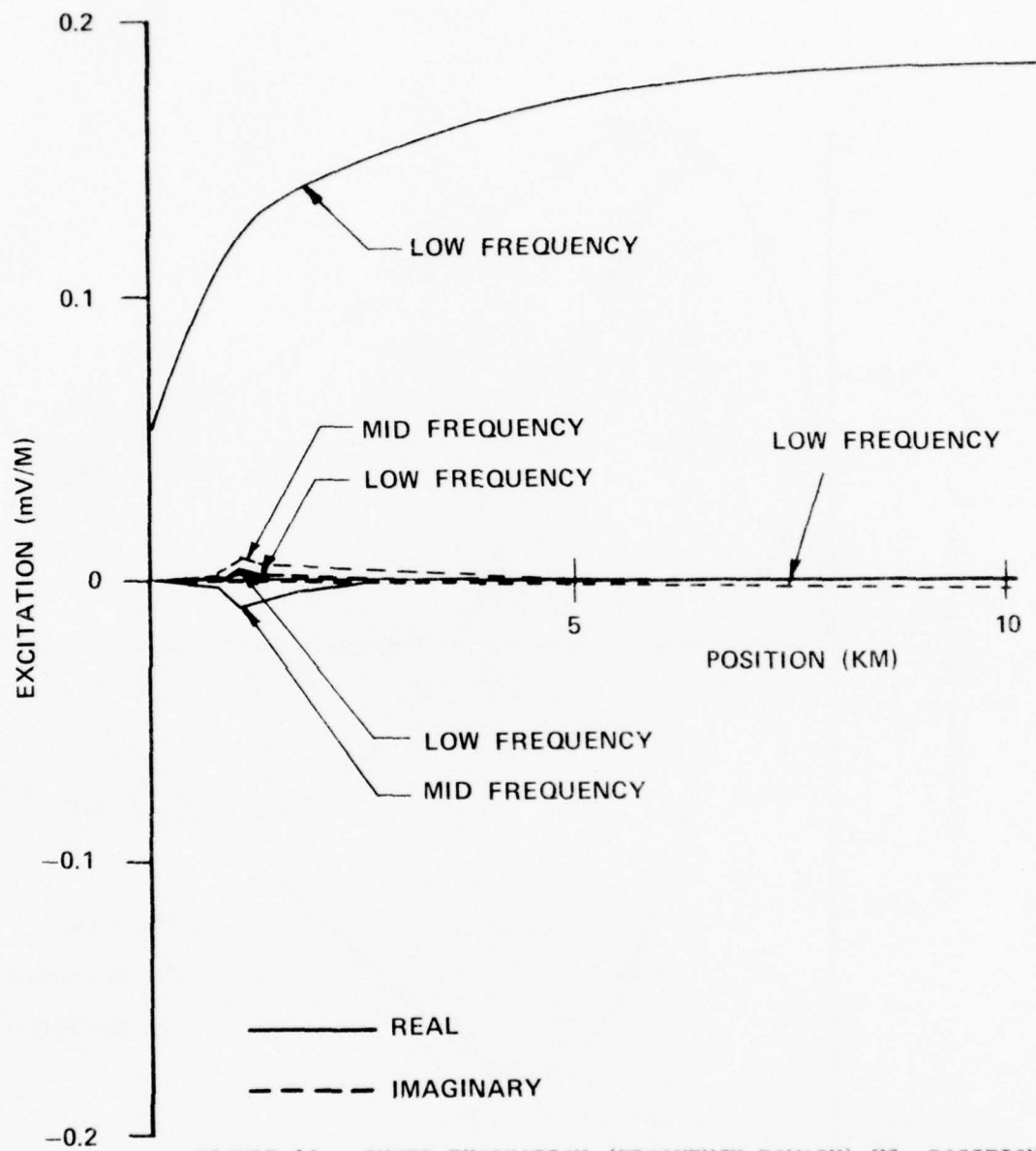


FIGURE 12. INNER EXCITATION (FREQUENCY DOMAIN) VS. POSITION

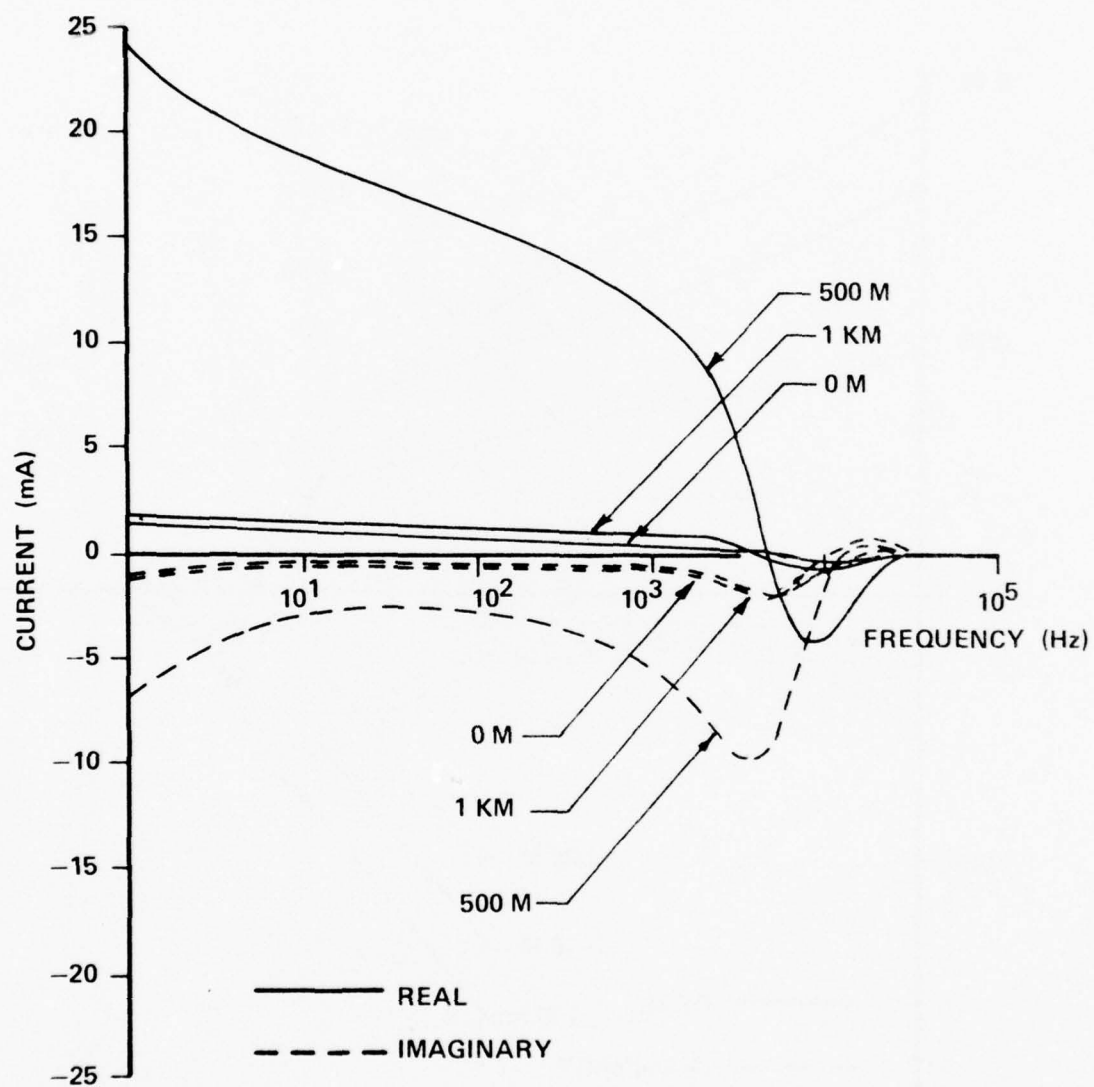


FIGURE 13. INNER CURRENT (FREQUENCY DOMAIN) VS. FREQUENCY

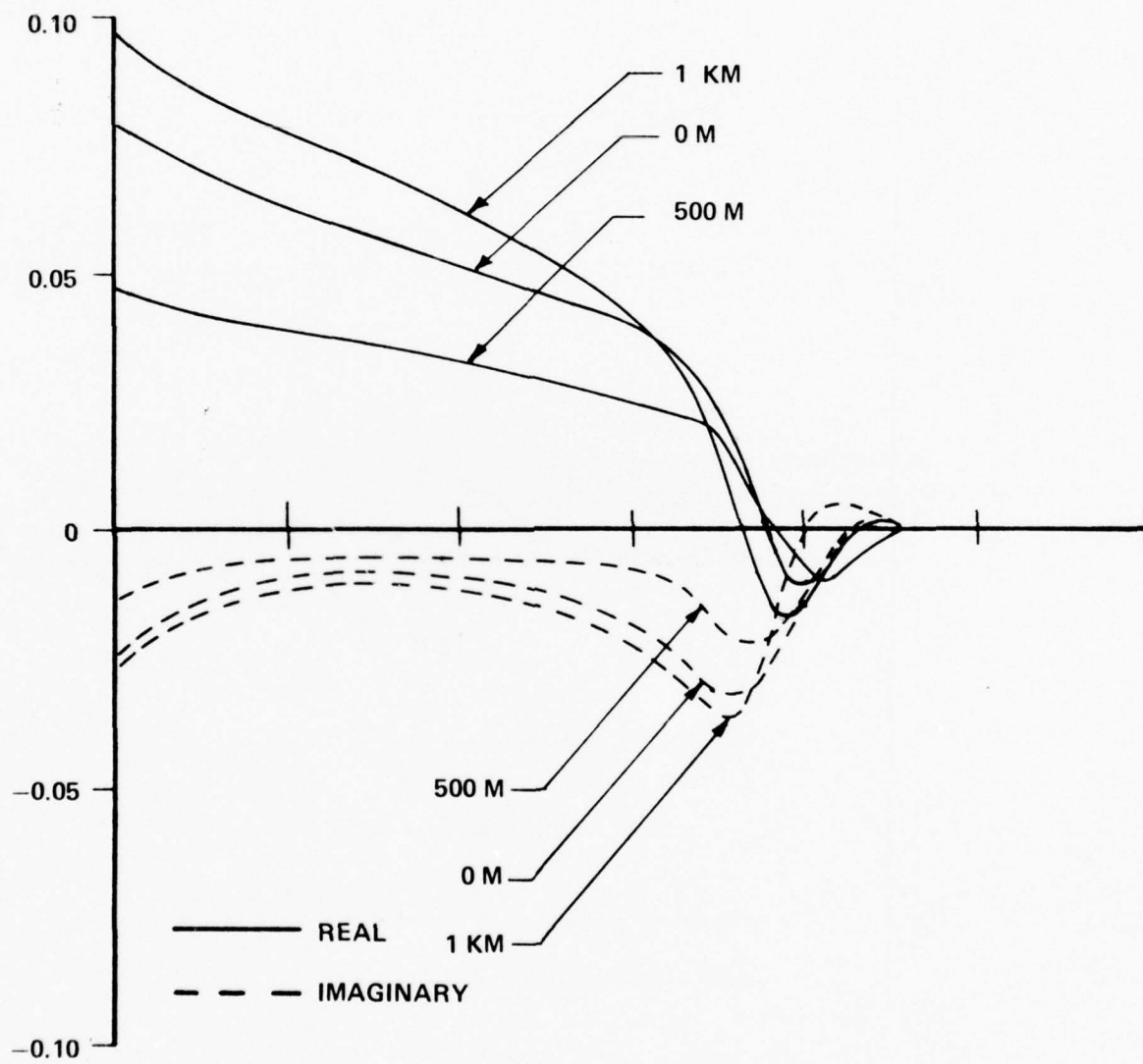


FIGURE 14. INNER VOLTAGE (FREQUENCY DOMAIN) VS. FREQUENCY

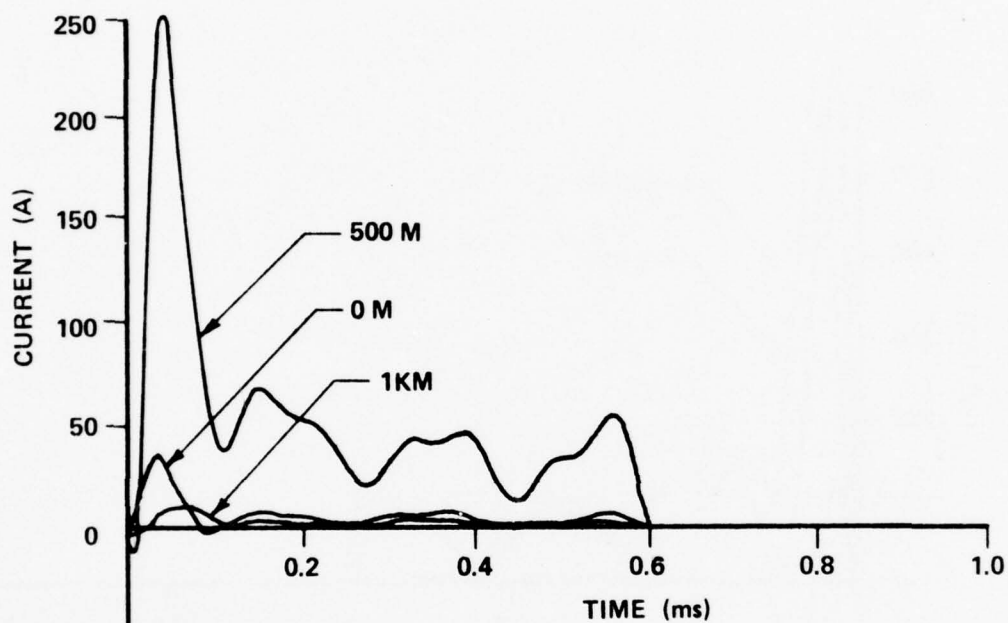


FIGURE 15. INNER CURRENT (TIME DOMAIN) VS. TIME

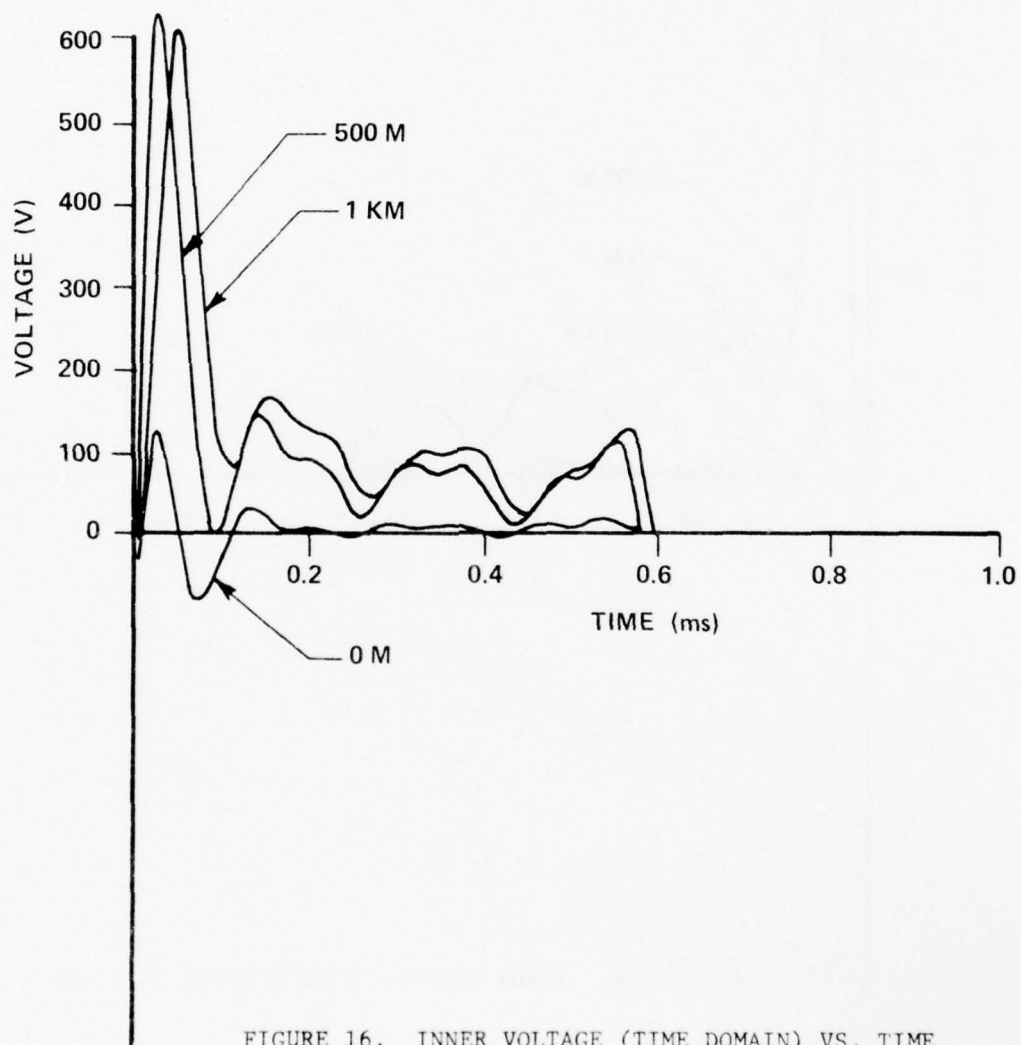


FIGURE 16. INNER VOLTAGE (TIME DOMAIN) VS. TIME

## APPENDIX A

PROGRAM LPS

Program Listing

```

+ (INPUT, OUTPUT, PLOTS,
+ DLSQ, DLSQ, DCOE, DCIE, DCOO, DCOV, DCIC, DCIV, DFTK, DFTU,
+ TAPE5=INPUT, TAPE6=OUTPUT, TAPE7=PLOTS,
+ TAPE21=DLSQ, TAPE22=DLSQ,
+ TAPE23=DCOE, TAPE24=DCIE,
+ TAPE25=DCOO, TAPE26=DCOV,
+ TAPE27=DCIC, TAPE28=DCIV,
+ TAPE29=DFTK, TAPE30=DFTU)

```

C#####

C MAIN PROGRAM

C#####

C FAA LIGHTNING PROTECTION STUDY

C CDC CYBER 74 (FORTRAN 4)

C GEORGIA TECH/SCHOOL OF EE/ATLANTA, GA 30532

C NORDDGARD / 12 MAY 76 (REVISED)

C

C PARAMETER DP=100

C PARAMETER DE=342, DX=32, DF=64, DT=128

C PARAMETER DA=130, CO=5

C PARAMETER DM=8

C PARAMETER DW=64

C

INTEGER DP

INTEGER DE, DX, DF, DT

INTEGER DA, DO

INTEGER DM

INTEGER DW

INTEGER CR, LP, PQ

INTEGER DLSQ, DLSQ

INTEGER DCOE, DCIE

INTEGER DCOO, DCOV

INTEGER DCIC, DCIV

INTEGER DFTK, DFTU

INTEGER RLS, RCO, FTT, PCI, RTL, RFT

INTEGER WLS, WCO, WTT, WCI, WTL, WFT

INTEGER TT, TX, TV, TH

REAL MIN, MAX

COMPLEX P, Q

COMPLEX K, C

COMPLEX U, V

COMPLEX E, IT

COMPLEX PC, ZC

COMPLEX MF

COMPLEX PCX, ZCX

COMPLEX YSX, ZSX

COMPLEX WNX, WIX

COMPLEX ZMX, ZPX

COMPLEX ZLM, ZLP

COMPLEX WJ, WH

COMPLEX JIWA, JIWW

COMPLEX JLIW, JLIW, YCIW, YLIW

COMPLEX JUIS, JUIS, YCIS, YLIS

COMPLEX HUES, HIES

COMPLEX EMW, EMI, EME

COMPLEX APCW, APCI, APCO

COMPLEX Y

C

DIMENSION P(51), Q(51)

DIMENSION E(41, 61), IT(41, 61)

DIMENSION K(3, 61), C(3, 61)

DIMENSION U(3, 61), V(3, 61)

DIMENSION PC(61), ZC(61)

DIMENSION MF(61)

A-1

DIMENSION A(63),O(63,5)  
 DIMENSION TT(8),TX(8),TV(6),TH(8)  
 DIMENSION X(61),Y(61)

C

COMMON/A10/CR,LP,PQ  
 COMMON/A11/DLSP,OLSD  
 COMMON/A12/DCOE,DCIE  
 COMMON/A13/DCOC,DCOV  
 COMMON/A14/DCIC,DCIV  
 COMMON/A15/DFTK,DFTU  
 COMMON/B11/IMX,EMX  
 COMMON/B12/ML,RL,SL  
 COMMON/B13/MIN,MAX,MD  
 COMMON/B14/NDX,MDX,MYX  
 COMMON/B15/NQ,NI  
 COMMON/B20/MN,NM  
 COMMON/B21/MLS,MCO,MTT,MCI,MTL,MFT  
 COMMON/B22/RLS,RCO,RTT,RCI,RTL,RFT  
 COMMON/B23/WLS,WCO,WTT,WCI,WTL,WFT  
 COMMON/B27/MRC  
 COMMON/B30/MQX  
 COMMON/B31/MM  
 COMMON/C11/TM,WM  
 COMMON/C12/NK,NC  
 COMMON/C13/TS,TR,TD,TP,TC  
 COMMON/C14/AX,BX,CX  
 COMMON/C15/APS,BFS,CPS  
 COMMON/C16/ARX,BRX,CRX  
 COMMON/C17/ARS,BRS,CRS  
 COMMON/C20/PGPS,PGRX,PGRS  
 COMMON/C21/PSPS,DCPS,DKPS,MPS,NPS  
 COMMON/C22/PSKX,DCRX,DKRX,MRX,NRX  
 COMMON/C23/PSRS,DCRS,DKRS,MRS,NRS  
 COMMON/C24/TSPS,TRPS,TPPS,TCPS  
 COMMON/C25/TSRX,TRRX,TDRX,TPRX,TCRX  
 COMMON/C26/TSRS,TRRS,TOKS,TPRS,TCRS  
 COMMON/D10/SI,SF,NS  
 COMMON/D11/XI,XF,NX  
 COMMON/D12/YI,YF,NY  
 COMMON/D13/WI,Wf,NW  
 COMMON/D14/TI,TF,NT  
 COMMON/D15/MS,MX,MY,MW,MT  
 COMMON/D16/IEI,IEF,NEI  
 COMMON/D17/IXI,IXF,NXI  
 COMMON/D18/IYI,IYF,NYI  
 COMMON/D19/IWI,IWF,NWI  
 COMMON/D20/ITI,ITF,NTI  
 COMMON/D21/XX,XT  
 COMMON/D22/XXI,XXF,NXX,MXX  
 COMMON/D23/XYI,XYF,NYX,MYX  
 COMMON/D24/XPI,XPY,NPX,MPX  
 COMMON/E11/RW  
 COMMON/E12/RSM,RSP  
 COMMON/E13/RAM,RAP  
 COMMON/E14/RC  
 COMMON/E15/ZO,YO  
 COMMON/F11/URE,ERE,UE,EE,OE  
 COMMON/F12/URI,ERI,UI,EI,OI  
 COMMON/F13/URA,ERA,UA,EA,OA  
 COMMON/F14/UR2,ER2,U2,E2,O2  
 COMMON/F15/URS,ERS,US,ES,OS  
 COMMON/F16/UR1,ER1,U1,E1,O1  
 COMMON/F17/URW,ERW,UW,EW,OW  
 COMMON/G11/ZLM,ZL

```

COMMON/H11/PCX,ZCX
COMMON/H12/YSX,ZSX
COMMON/H13/WNX,WIX
COMMON/H14/ZMX,ZPX
COMMON/I10/WJ,WH
COMMON/I11/J0WW,J1WW
COMMON/I12/J0IW,J1IW,Y0IW,Y1IW
COMMON/I13/J0IS,J1IS,Y0IS,Y1IS
COMMON/I14/H0ES,H1ES
COMMON/I15/EMW,EMI,EME
COMMON/I16/APCW,AFCI,APCE

C
C EQUIVALENCE (K,C)
C EQUIVALENCE (U,V)
C

DATA CR,LP,PQ/5,6,7/
DATA DLSP,DLSO/21,22/
DATA DCOE,DCIE/23,24/
DATA DCOO,DOOV/25,26/
DATA DCIC,DCIV/27,28/
DATA DFTK,DFTU/29,30/
DATA DP/51/
DATA DE,CX,DF,DT/41,3,61,61/
DATA DA,DO/63,5/
DATA DM/6/
DATA DW/61/

C
CALL RS
+(A,O,DA,DO,TT,TX,TV,TH,DM,X,Y,DW,
+ P,Q,DP,K,C,U,V,E,IT,PC,ZC,MR,DE,DX,DF,DT)

C
STOP
END
C#####
C RUNSTREAM
C#####
SUBROUTINE RS
+(A,O,PA,PO,TT,TX,TV,TH,PM,X,Y,PW,
+ P,Q,PP,K,C,U,V,E,IT,PC,ZC,MR,PE,PX,PF,PT)

C
C RUNSTREAM
C
INTEGER PP
INTEGER PE,PX,PF,PT
INTEGER PA,PO
INTEGER PM
INTEGER PW
INTEGER TT,TX,TV,TH
COMPLEX P,Q
COMPLEX K,C
COMPLEX U,V
COMPLEX E,IT
COMPLEX PC,ZC
COMPLEX MR
COMPLEX Y

C
DIMENSION P(PF),Q(PF)
DIMENSION K(PX,PT),C(PX,PF)
DIMENSION U(PX,PT),V(PX,PF)
DIMENSION E(PE,PF),IT(PE,PF)
DIMENSION PC(PF),ZC(PF)
DIMENSION MR(PF)
DIMENSION A(PA),O(PA,PO)
DIMENSION TT(PM),TX(PM),TV(PM),TH(PM)

```

```

      DIMENSION X(PW),Y(PW)
C
      COMMON/B11/MLS,MCC,MTT,MCI,MTL,MFT
C
      CALL PSIO
C
      IF(MLS.NE.0) CALL PSLS
      +(A,O,PA,PO,TT,TX,TV,TH,PM,P,Q,PP)
      IF(MCC.NE.0) CALL PSCC
      +(A,O,PA,PO,TT,TX,TV,TH,PM,E,PC,ZC,PE,PF)
      IF(MTL.NE.0) CALL PSTL
      +(A,O,PA,PO,TT,TX,TV,TH,PM,X,Y,PW,IT,IT,E,PC,ZC,PE,PE,PF)
      IF(MTT.NE.0) CALL PST
      +(A,O,PA,PO,TT,TX,TV,TH,PM,MR,PF)
      IF(MCI.NE.0) CALL PSCI
      +(A,O,PA,PO,TT,TX,TV,TH,PM,E,PC,ZC,IT,MR,PE,PF)
      IF(MTL.NE.0) CALL PSTL
      +(A,O,PA,PO,TT,TX,TV,TH,PM,X,Y,PW,C,V,E,PC,ZC,PE,PX,PF)
      IF(MFT.NE.0) CALL PSFT
      +(A,O,PA,PO,TT,TX,TV,TH,PM,X,Y,PW,K,C,U,V,PX,PF,PT)
C
      RETURN
      END
C#####
C      PROGRAM SEQUENCES
C#####
      SUBROUTINE PSIO
C
      INPUT/OUTPUT
C
      INTEGER CP,LP,PQ
      INTEGER RLS,RCO,RTT,PCI,RTL,RFT
      INTEGER WLS,WCO,WTT,WCI,WTL,WFT
      REAL    MIN,MAX
      COMPLEX ZLM,ZLP
C
      COMMON/A10/CP,LP,PQ
      COMMON/B11/IMX,EMX
      COMMON/B12/ML,RL,SL
      COMMON/B13/MIN,MAX,MP
      COMMON/B14/NOX,MOX,MXX
      COMMON/B15/NQ,NI
      COMMON/B20/MN,NM
      COMMON/B21/MLS,MCC,MTT,MCI,MTL,MFT
      COMMON/B22/RLS,RCO,RTT,PCI,RTL,RFT
      COMMON/B23/WLS,WCO,WTT,WCI,WTL,WFT
      COMMON/B27/MP
      COMMON/B30/MPX
      COMMON/C11/TM,WM
      COMMON/C12/NK,NC
      COMMON/C20/PGRX,PGES
      COMMON/C21/PSPS,DCPS,DKPS,MPS,NPS
      COMMON/C22/PSRX,DCRX,DKRX,MFX,NRX
      COMMON/C23/PSRS,DCRS,DKFS,MFS,NRS
      COMMON/C24/TSPS,TRPS,TDFS,TAPS,TCFS
      COMMON/C25/TSPX,TRRX,TDRX,TPRX,TCRX
      COMMON/C26/TSKS,TRRS,TDRS,TPRS,TCRS
      COMMON/D10/SI,SF,NS
      COMMON/D11/XI,XF,NX
      COMMON/D12/YI,YF,NY
      COMMON/D13/WI,WF,NW
      COMMON/D14/II,IF,NI
      COMMON/D15/MS,MX,MY,MW,MT
      COMMON/D16/IEI,IEF,NEI

```

```

COMMON/D17/IXI,IXF,IXI
COMMON/D18/IYI,IYF,NIYI
COMMON/D19/IWI,IWF,NWI
COMMON/D20/ITI,ITF,NTI
COMMON/E11/RW
COMMON/E12/RSM,RSP
COMMON/E13/RAH,RAF
COMMON/E14/RS
COMMON/E15/DO,SO
COMMON/F11/JRE,ERE,UE,EE,OE
COMMON/F12/JRI,ERI,UI,EI,OI
COMMON/F13/JRA,ERA,UA,EA,OA
COMMON/F14/JR2,ER2,U2,E2,O2
COMMON/F15/JRS,ERS,US,ES,OS
COMMON/F16/JRI,ERI,UI,EI,OI
COMMON/F17/JRW,EPW,UW,EW,OW
COMMON/G11/ZLM,ZLP

```

C

```

1  FORMAT(10I10)
2  FORMAT(10G10.0)
3  FORMAT(1I10,10G10.0)
4  FORMAT(2G10.0,10I10)
5  FORMAT(3G10.0,10I10)
10 FORMAT("1","INPUT/OUTPUT")
11 FORMAT("0","MAXIMUM ITERATION",I20 /
+      " ", "MAXIMUM ERROR",1PG20.10)
12 FORMAT("0","LOG MODE",I20 /
+      " ", "LOG RATIO",1PG20.10/
+      " ", "LOG SCALE",1PG20.10)
13 FORMAT("0","PLOT MINIMUM",1PG20.10/
+      " ", "PLOT MAXIMUM",1PG20.10/
+      " ", "PLOT MODE",I20 )
14 FORMAT("0","POLYNOMIAL DEGREE",I20 /
+      " ", "CALCULATE DERIVATIVE",I20 /
+      " ", "TABLE ORDER",I20 )
15 FORMAT("0","QUADRATURE GRID",I20 /
+      " ", "INTEGRATION GRID",I20 )
16 FORMAT(" ", "TITLE NUMBER",I20 )
21 FORMAT("0","MODE STROKE",I20 /
+      " ", "MODE COUPLE (OUTER)",I20 /
+      " ", "MODE TRANSFER",I20 /
+      " ", "MODE COUPLE (INNER)",I20 /
+      " ", "MODE TRANSMISSION",I20 /
+      " ", "MODE INVERSE",I20 )
22 FORMAT("0","READ STROKE",I20 /
+      " ", "READ COUPLE (OUTER)",I20 /
+      " ", "READ TRANSFER",I20 /
+      " ", "READ COUPLE (INNER)",I20 /
+      " ", "READ TRANSMISSION",I20 /
+      " ", "READ INVERSE",I20 )
23 FORMAT("0","WRITE STROKE",I20 /
+      " ", "WRITE COUPLE (OUTER)",I20 /
+      " ", "WRITE TRANSFER",I20 /
+      " ", "WRITE COUPLE (INNER)",I20 /
+      " ", "WRITE TRANSMISSION",I20 /
+      " ", "WRITE INVERSE",I20 )
27 FORMAT("0","MODE ROOT",I20 )
30 FORMAT(" ", "MODE PLOT",I20 )
31 FORMAT("0","MULTIPLIER",1PG20.10,1X,"TIME DOMAIN"/
+      " ", "MULTIPLIER",1PG20.10,1X,"FREQUENCY DOMAIN"/
32 FORMAT("0","NUMBER OF PULSES",I20 ,1X,"TIME DOMAIN"/
+      " ", "NUMBER OF PULSES",I20 ,1X,"FREQUENCY DOMAIN"/
40 FORMAT("0","PULSE GUESS PRESTRIKE",1PG20.10/
+      " ", "PULSE GUESS RETURN",1PG20.10/

```

A-5

```

+      " ", "PULSE GUESS      RESTRIKE      ", 1PG20.10)
41 FORMAT("0", "PULSE PEAK    PRESTRIKE      ", 1PG20.10/
+      " ", "DECAY CONSTANT  PRESTRIKE      ", 1PG20.10/
+      " ", "DECAY CONSTANT  PRESTRIKE      ", 1PG20.10/
+      " ", "MODE DECAY      PRESTRIKE      ", I20 /
+      " ", "MODE NUMBER     PRESTRIKE      ", I20 )
42 FORMAT("0", "PULSE PEAK    RETURN         ", 1PG20.10/
+      " ", "DECAY CONSTANT  RETURN         ", 1PG20.10/
+      " ", "DECAY CONSTANT  RETURN         ", 1PG20.10/
+      " ", "MODE DECAY      RETURN         ", I20 /
+      " ", "MODE NUMBER     RETURN         ", I20 )
43 FORMAT("0", "PULSE PEAK    RESTRIKE      ", 1PG20.10/
+      " ", "DECAY CONSTANT  RESTRIKE      ", 1PG20.10/
+      " ", "DECAY CONSTANT  RESTRIKE      ", 1PG20.10/
+      " ", "MODE DECAY      RESTRIKE      ", I20 /
+      " ", "MODE NUMBER     RESTRIKE      ", I20 )
44 FORMAT("0", "TIME START    PRESTRIKE      ", 1PG20.10/
+      " ", "TIME RISE       PRESTRIKE      ", 1PG20.10/
+      " ", "TIME DECAY      PRESTRIKE      ", 1PG20.10/
+      " ", "TIME PULSE      PRESTRIKE      ", 1PG20.10/
+      " ", "TIME CONSTANT   PRESTRIKE      ", 1PG20.10)
45 FORMAT("0", "TIME START    RETURN         ", 1PG20.10/
+      " ", "TIME RISE       RETURN         ", 1PG20.10/
+      " ", "TIME DECAY      RETURN         ", 1PG20.10/
+      " ", "TIME PULSE      RETURN         ", 1PG20.10/
+      " ", "TIME CONSTANT   RETURN         ", 1PG20.10)
46 FORMAT("0", "TIME START    RESTRIKE      ", 1PG20.10/
+      " ", "TIME RISE       RESTRIKE      ", 1PG20.10/
+      " ", "TIME DECAY      RESTRIKE      ", 1PG20.10/
+      " ", "TIME PULSE      RESTRIKE      ", 1PG20.10/
+      " ", "TIME CONSTANT   RESTRIKE      ", 1PG20.10)
50 FORMAT("0", "INITIAL EXCITATION ", 1PG20.10/
+      " ", "FINAL EXCITATION  ", 1PG20.10/
+      " ", "NUMBER OF BANDS   ", I20 )
51 FORMAT("0", "INITIAL POSITION ", 1PG20.10/
+      " ", "FINAL POSITION     ", 1PG20.10/
+      " ", "NUMBER OF BANDS   ", I20 )
52 FORMAT("0", "INITIAL FREQUENCY ", 1PG20.10/
+      " ", "FINAL FREQUENCY   ", 1PG20.10/
+      " ", "NUMBER OF BANDS   ", I20 )
53 FORMAT("0", "INITIAL TIME     ", 1PG20.10/
+      " ", "FINAL TIME        ", 1PG20.10/
+      " ", "NUMBER OF BANDS   ", I20 )
54 FORMAT("0", "MODE EXCITATION ", I20 /
+      " ", "MODE POSITION     ", I20 /
+      " ", "MODE POSITION     ", I20 /
+      " ", "MODE FREQUENCY   ", I20 /
+      " ", "MODE TIME        ", I20 )
55 FORMAT("0", "INITIAL EXCITATION ", I20 /
+      " ", "FINAL EXCITATION  ", I20 /
+      " ", "NUMBER OF SAMPLES ", I20 )
56 FORMAT("0", "INITIAL POSITION ", I20 /
+      " ", "FINAL POSITION     ", I20 /
+      " ", "NUMBER OF SAMPLES ", I20 )
57 FORMAT("0", "INITIAL FREQUENCY ", I20 /
+      " ", "FINAL FREQUENCY   ", I20 /
+      " ", "NUMBER OF SAMPLES ", I20 )
58 FORMAT("0", "INITIAL TIME     ", I20 /
+      " ", "FINAL TIME        ", I20 /
+      " ", "NUMBER OF SAMPLES ", I20 )
61 FORMAT("0", "RADIUS WIRE     ", 1PG20.10)
62 FORMAT(" ", "RADIUS SHIELD MINUS ", 1PG20.10/
+      " ", "RADIUS SHIELD PLUS  ", 1PG20.10)
63 FORMAT(" ", "RADIUS ARMOR MINUS ", 1PG20.10/

```

```

+      " ", "RADIUS ARMOR PLUS ", 1PG20.10)
64 FORMAT(" ", "RADIUS SHEATH ", 1PG20.10)
65 FORMAT("0", "DEPTH ", 1PG20.10/
+      " ", "SEPARATION ", 1PG20.10)
71 FORMAT("0", "RELATIVE PERMEABILITY EARTH ", 1PG20.10/
+      " ", "RELATIVE PERMITTIVITY EARTH ", 1PG20.10/
+      " ", "ABSOLUTE PERMEABILITY EARTH ", 1PG20.10/
+      " ", "ABSOLUTE PERMITTIVITY EARTH ", 1PG20.10/
+      " ", "ABSOLUTE CONDUCTIVITY EARTH ", 1PG20.10)
72 FORMAT("0", "RELATIVE PERMEABILITY SHEATH ", 1PG20.10/
+      " ", "RELATIVE PERMITTIVITY SHEATH ", 1PG20.10/
+      " ", "ABSOLUTE PERMEABILITY SHEATH ", 1PG20.10/
+      " ", "ABSOLUTE PERMITTIVITY SHEATH ", 1PG20.10/
+      " ", "ABSOLUTE CONDUCTIVITY SHEATH ", 1PG20.10)
73 FORMAT("0", "RELATIVE PERMEABILITY ARMOR ", 1PG20.10/
+      " ", "RELATIVE PERMITTIVITY ARMOR ", 1PG20.10/
+      " ", "ABSOLUTE PERMEABILITY ARMOR ", 1PG20.10/
+      " ", "ABSOLUTE PERMITTIVITY ARMOR ", 1PG20.10/
+      " ", "ABSOLUTE CONDUCTIVITY ARMOR ", 1PG20.10)
74 FORMAT("0", "RELATIVE PERMEABILITY 2 ", 1PG20.10/
+      " ", "RELATIVE PERMITTIVITY 2 ", 1PG20.10/
+      " ", "ABSOLUTE PERMEABILITY 2 ", 1PG20.10/
+      " ", "ABSOLUTE PERMITTIVITY 2 ", 1PG20.10/
+      " ", "ABSOLUTE CONDUCTIVITY 2 ", 1PG20.10)
75 FORMAT("0", "RELATIVE PERMEABILITY SHIELD ", 1PG20.10/
+      " ", "RELATIVE PERMITTIVITY SHIELD ", 1PG20.10/
+      " ", "ABSOLUTE PERMEABILITY SHIELD ", 1PG20.10/
+      " ", "ABSOLUTE PERMITTIVITY SHIELD ", 1PG20.10/
+      " ", "ABSOLUTE CONDUCTIVITY SHIELD ", 1PG20.10)
76 FORMAT("0", "RELATIVE PERMEABILITY 1 ", 1PG20.10/
+      " ", "RELATIVE PERMITTIVITY 1 ", 1PG20.10/
+      " ", "ABSOLUTE PERMEABILITY 1 ", 1PG20.10/
+      " ", "ABSOLUTE PERMITTIVITY 1 ", 1PG20.10/
+      " ", "ABSOLUTE CONDUCTIVITY 1 ", 1PG20.10)
77 FORMAT("0", "RELATIVE PERMEABILITY WIRE ", 1PG20.10/
+      " ", "RELATIVE PERMITTIVITY WIRE ", 1PG20.10/
+      " ", "ABSOLUTE PERMEABILITY WIRE ", 1PG20.10/
+      " ", "ABSOLUTE PERMITTIVITY WIRE ", 1PG20.10/
+      " ", "ABSOLUTE CONDUCTIVITY WIRE ", 1PG20.10)
81 FORMAT("0", "LOAD IMPEDANCE MINUS", 2(1PG20.10)/
+      " ", "LOAD IMPEDANCE PLUS ", 2(1PG20.10))

```

C

```

READ(CR, 3) IMX, EMX
READ(CR, 3) ML, RL, SL
READ(CR, 4) MIN, MAX, MP
READ(CR, 1) NDX, MCX, MXX
READ(CR, 1) NO, NI
READ(CR, 1) MN, NM
READ(CR, 1) MLS, MCO, MIT, MCI, MTL, MFT
READ(CR, 1) RLS, FCO, RTT, FCI, RTL, RFT
READ(CR, 1) WLS, WCO, WTT, WCI, WTL, WFT
READ(CR, 1) MR
READ(CR, 1) MFX
READ(CR, 2) TM, WM
READ(CR, 1) NK, NC
READ(CR, 2) PGRS, PGRX, PGRS
READ(CR, 5) PSPS, DCPS, DKPS, MPS, NPS
READ(CR, 5) PSRX, DCRX, DKRX, MFX, NFX
READ(CR, 5) PSRS, DCRS, DKRS, MFS, NRS
READ(CR, 2) TSPS, TRPS, TOPS, TPRS, TCPS
READ(CR, 2) TSRX, TRRX, TORX, TPRX, TCRX
READ(CR, 2) TSRS, TRRS, TDRS, TRRS, TCRS
READ(CR, 4) SI, SF, NS
READ(CR, 4) XI, XF, NX

```

```

READ(CR, 4) YI,YF,NY
READ(CR, 4) WI,Wf,NW
READ(CR, 4) TI,TF,NT
READ(CR, 1) IS,IX,MY,MW,MT
READ(CR, 1) ICI,IEF,NEI
READ(CR, 1) IXI,IXF,NXI
READ(CR, 1) IYI,IYF,NYI
READ(CR, 1) IWI,IWF,NWI
READ(CR, 1) ITI,ITF,NTI
READ(CR, 2) RW
READ(CR, 2) RSM,RSP
READ(CR, 2) RAM,RAP
READ(CR, 2) RS
READ(CR, 2) DO
READ(CR, 2) SO
READ(CR, 2) URE,ERE,OE
READ(CR, 2) UKI,ERI,OI
READ(CR, 2) UPA,ERA,OA
READ(CR, 2) U-2,ER2,O2
READ(CR, 2) UKS,ERS,OS
READ(CR, 2) UK1,ER1,O1
READ(CR, 2) UPW,EPW,OW
READ(CR, 2) ZLM
READ(CR, 2) ZLP

```

C

```
CALL SRC
```

C

```
READ(CR, 1) MID IF(MID.LE.0) RETURN
```

C

```

WRITE(LP,10)
WRITE(LP,11) IMX,EMX
WRITE(LP,12) ML,PL,SL
WRITE(LP,13) MIN,MAX,MP
WRITE(LP,14) NOX,MOX,MXX
WRITE(LP,15) NO,NI
WRITE(LP,16) MN
WRITE(LP,16) NM
WRITE(LP,21) MLS,MCO,MTT,MOI,MTL,MFT
WRITE(LP,22) RLS,RCO,RTT,ROI,RTL,FFT
WRITE(LP,23) WLS,WCO,WTT,WCI,WTL,WFT
WRITE(LP,27) MR
WRITE(LP,30) MPX
WRITE(LP,31) TM,WM
WRITE(LP,32) NK,NC
WRITE(LP,40) PGPS,PGRX,EGRS
WRITE(LP,41) PSFS,DCPS,DKFS,MPS,NPS
WRITE(LP,42) PSRX,DCRX,DKRX,MXX,NPX
WRITE(LP,43) PSFS,DCRS,UKRS,MRS,NRS
WRITE(LP,44) TSFS,TRFS,TDFS,TPPS,TCFS
WRITE(LP,45) TSRX,TRRX,TDRX,TPRX,TCRX
WRITE(LP,46) TSPS,TRRS,TDRS,TPRS,TCRS
WRITE(LP,50) SI,SF,NS
WRITE(LP,51) XI,XF,NX
WRITE(LP,51) YI,YF,NY
WRITE(LP,52) WI,Wf,NW
WRITE(LP,53) TI,TF,NT
WRITE(LP,54) MS,MX,MY,MW,MT
WRITE(LP,55) ICI,IEF,NEI
WRITE(LP,56) IXI,IXF,NXI
WRITE(LP,56) IYI,IYF,NYI
WRITE(LP,57) IWI,IWF,NWI
WRITE(LP,58) ITI,ITF,NTI
WRITE(LP,61) RW
WRITE(LP,62) RSM,RSP

```

```

WRITE(LP,63) RAM,RAP
WRITE(LP,64) RS
WRITE(LP,65) DO,SO
WRITE(LP,71) URE,ERE,UE,EE,OE
WRITE(LP,72) URI,ERI,UI,EI,OI
WRITE(LP,73) URA,ERA,UA,EA,OA
WRITE(LP,74) UR2,ER2,U2,E2,O2
WRITE(LP,75) URS,ERS,US,ES,OS
WRITE(LP,76) UR1,ER1,U1,E1,O1
WRITE(LP,77) URW,ERW,UW,EW,OW
WRITE(LP,81) ZLM,ZLP

C
RETURN
END
C*****
SUBROUTINE PSLS(A,O,PA,PO,TT,TX,TV,TH,PM,P,Q,PS)
C
C LIGHTNING STROKE
C
INTEGER PS
INTEGER PA,PO
INTEGER PM
INTEGER CR,LP,PQ
INTEGER TT,TX,TV,TH
COMPLEX P,Q

C
DIMENSION P(PS),Q(PS)
DIMENSION A(PA),O(PA,PO)
DIMENSION TT(PM),TX(PM),TV(PM),TH(PM)

C
COMMON/A10/CR,LP,PQ
COMMON/C11/TM,WM
COMMON/C12/NK,NC
COMMON/D15/ME,MX,MY,MW,MT

C 10 FORMAT("1","LIGHTNING STROKE")
C
DATA N1/1/
DATA TO,WO/U.O,1.O/

C
WRITE(LP,10) $CALL SRSC $CALL SFLS(P,Q,PS)

C
CALL TPGX(A,O,PA,PO,TT,TX,TV,TH,PM,P,PS,TO,TM,NK,MT,N1,NK,N1)
CALL TPGX(A,O,PA,PO,TT,TX,TV,TH,PM,Q,PS,WO,WM,NC,MW,N1,NC,N1)

C
RETURN
END
C*****
SUBROUTINE PSCU(A,O,PA,PO,TT,TX,TV,TH,PM,E,PC,ZC,PE,PF)
C
C COUPLING
C
INTEGER PE,PF
INTEGER PA,PO
INTEGER PM
INTEGER CR,LP,PQ
INTEGER DCOE,DCIE
INTEGER RLS,RCO,RTT,RCI,RTL,FFT
INTEGER WLS,WCO,WTT,WCI,WTL,WFT
INTEGER TT,TX,TV,TH
COMPLEX E
COMPLEX PC,ZC

C
DIMENSION E(PE,PF)

```

```

      DIMENSION PC(PF),ZC(PF)
      DIMENSION A(PA),O(PA,PO)
      DIMENSION TT(PM),TX(PM),TV(PM),TH(PM)
C
      COMMON/A10/CR,LP,PQ
      COMMON/A12/DOOE,DOIE
      COMMON/B22/RLS,RCO,RTT,RCI,RTL,RFT
      COMMON/B23/WLS,WCO,WTT,WCI,WTL,WFT
      COMMON/B31/MM
      COMMON/D10/EI,EF,NE
      COMMON/D13/WI,WF,NW
      COMMON/D15/ME,MX,MY,MW,MT
      COMMON/D16/IE1,IEF,NEI
      COMMON/D19/IWI,IWF,NWI
C
      11 FORMAT("1","PROPAGATION PARAMETERS")
      12 FORMAT("1","COUPLING (OUTER)")
C
      MM=1
C
      WRITE(LP,11) $CALL SRS(E,PC,ZC,PE,PF)
      WRITE(LP,12)
      CALL TPGK
      +(A,O,PA,PO,TT,TX,TV,TH,PM,E,PE,PF,
      + EI,EF,NE,ME,IE1,IEF,NEI,WI,WF,NW,MW,IWI,IWF,NWI)
C
      IF(WCO.NE.0) CALL FTOW(DOOE,E,PE,PF)
C
      RETURN
      END
C*****
      SUBROUTINE PST(A,O,PA,PO,TT,TX,TV,TH,PM,MR,PF)
C
C      TRANSFER (OUTER/INNER)
C
      INTEGER PF
      INTEGER PA,PO
      INTEGER PM
      INTEGER CR,LP,PQ
      INTEGER TT,TX,TV,TH
      COMPLEX MR
C
      DIMENSION MR(PF)
      DIMENSION A(PA),O(PA,PO)
      DIMENSION TT(PM),TX(PM),TV(PM),TH(PM)
C
      COMMON/A10/CR,LP,PQ
      COMMON/D13/WI,WF,NW
      COMMON/D15/ME,MX,MY,MW,MT
C
      10 FORMAT("1","TRANSFER (OUTER/INNER)")
C
      DATA N1/1/
C
      WRITE(LP,10) $CALL SRT(MR,PF)
C
      CALL TPGX(A,O,PA,PO,TT,TX,TV,TH,PM,MR,PF,WI,WF,NW,MW,N1,NW,N1)
C
      RETURN
      END
C*****
      SUBROUTINE PSCI(A,O,PA,PO,TT,TX,TV,TH,PM,E,PC,ZC,IT,MR,PE,PF)
C
C      COUPLING

```

```

C      INTEGER PE,PF
      INTEGER PA,PO
      INTEGER PM
      INTEGER CR,LP,PQ
      INTEGER DCOE,DCIE
      INTEGER RLS,RCO,RTT,RCI,RTL,RFT
      INTEGER WLS,WCO,WTT,WCI,WTL,WFT
      INTEGER TT,TX,TV,TH
      COMPLEX E,IT
      COMPLEX PC,ZC
      COMPLEX MR

C      DIMENSION E(PE,PF),IT(PE,PF)
      DIMENSION PC(PF),ZC(PF)
      DIMENSION MR(PF)
      DIMENSION A(PA),O(PA,PO)
      DIMENSION TT(PM),TX(PM),TV(PM),TH(PM)

C      COMMON/A10/CR,LP,PQ
      COMMON/A12/DCOE,DCIE
      COMMON/B22/RLS,RCO,RTT,RCI,RTL,RFT
      COMMON/B23/WLS,WCO,WTT,WCI,WTL,WFT
      COMMON/B31/MM
      COMMON/D11/XI,XF,NX
      COMMON/D13/WI,WF,NW
      COMMON/D15/ME,MX,MY,MW,MT
      COMMON/D17/IXI,IXF,NXI
      COMMON/D19/IWI,IWF,NWI

C      11 FORMAT("1","PROPAGATION PARAMETERS")
      12 FORMAT("1","COUPLING (INNER)")

C      MM=0

C      WRITE(LP,11) $CALL SRCC(E,PC,ZC,IT,MR,PE,PX,PF)
      WRITE(LP,12)
      CALL TPGK
      +(A,O,PA,PO,TT,TX,TV,TH,PM,E,PE,PF,
      + XI,XF,NX,MX,IXI,IXF,NXI,WI,WF,NW,MW,IWI,IWF,NWI)

C      IF(WCI.NE.0) CALL FTGW(DCIE,E,PE,PF)

C      RETURN
      END

C*****
C      SUBROUTINE PSTL
      +(A,O,PA,PO,TT,IX,TV,TH,PM,X,Y,PW,C,V,E,PC,ZC,PE,PX,PF)

C      TRANSMISSION LINE

C      INTEGER PE,PX,PF
      INTEGER PA,PO
      INTEGER PM
      INTEGER PW
      INTEGER CR,LP,PQ
      INTEGER DCOE,DCIE
      INTEGER DCOO,DCOV
      INTEGER DCIC,DCIV
      INTEGER RLS,RCO,RTT,RCI,RTL,RFT
      INTEGER WLS,WCO,WTT,WCI,WTL,WFT
      INTEGER TT,TX,TV,TH
      COMPLEX C,V
      COMPLEX E

```

```

COMPLEX PC,ZC
COMPLEX Y
C
  DIMENSION C(PX,PF),V(PX,PF)
  DIMENSION L(PE,PF)
  DIMENSION PC(PF),ZC(PF)
  DIMENSION A(PA),O(PA,PQ)
  DIMENSION TT(PM),TX(PM),TV(PM),TH(PM)
  DIMENSION X(PW),Y(PW)
C
  COMMON/A10/CR,LP,PQ
  COMMON/A12/DCOE,DCIE
  COMMON/A13/DCOC,DCOV
  COMMON/A14/DCIC,DCIV
  COMMON/B22/RLS,RCO,RTT,RCI,RTL,RFT
  COMMON/B23/WLS,WCO,WTT,WCI,WTL,WFT
  COMMON/B31/MM
  COMMON/D11/XI,XF,NX
  COMMON/D12/YI,YF,NY
  COMMON/D13/WI,WF,NW
  COMMON/D15/MI,MX,MY,MW,MT
  COMMON/D17/IXI,IXF,NXI
  COMMON/D18/IYI,IYF,NYI
  COMMON/D19/IWI,IWF,NWI
C
  10 FORMAT("1","TRANSMISSION LINE")
C
  IF((RTL.NE.0).AND.(MM.EQ.1)) CALL FTDR(DCOE,E,PE,PF)
  IF((RTL.NE.0).AND.(MM.EQ.0)) CALL FTDR(DCIE,E,PE,PF)
C
  WRITE(LP,10) 3CALL SRTL(X,Y,PW,C,V,E,PC,ZC,PE,PX,PF)
C
  IF(MM.EQ.1) CALL TPGK
  +(A,O,PA,PQ,TT,TX,TV,TH,PM,C,PX,PF,
  + XI,XF,NX,MX,IXI,IXF,NXI,WI,WF,NW,MW,IWI,IWF,NWI)
  IF(MM.EQ.0) CALL TPGK
  +(A,O,PA,PQ,TT,TX,TV,TH,PM,C,PX,PF,
  + YI,YF,NY,MY,IYI,IYF,NYI,WI,WF,NW,MW,IWI,IWF,NWI)
  IF(MM.EQ.0) CALL TPGK
  +(A,O,PA,PQ,TT,TX,TV,TH,PM,V,PX,PF,
  + YI,YF,NY,MY,IYI,IYF,NYI,WI,WF,NW,MW,IWI,IWF,NWI)
C
  IF((WTL.NE.0).AND.(MM.EQ.1)) CALL FTDW(DCOC,C,PX,PF)
  IF((WTL.NE.0).AND.(MM.EQ.0)) CALL FTDW(DCIC,C,PX,PF)
  IF((WTL.NE.0).AND.(MM.EQ.0)) CALL FTDW(DCIV,V,PX,PF)
C
  RETURN
  END
C*****
  SUBROUTINE PSFT
  +(A,O,PA,PQ,TT,TX,TV,TH,PM,X,Y,PW,K,C,U,V,PX,PF,PT)
C
  FOURIER TRANSFORM
C
  INTEGER PX,PF,PT
  INTEGER PA,PQ
  INTEGER PM
  INTEGER PW
  INTEGER CR,LP,PQ
  INTEGER DCIC,DCIV
  INTEGER DFTK,DFTU
  INTEGER RLS,RCO,RTT,RCI,RTL,RFT
  INTEGER WLS,WCO,WTT,WCI,WTL,WFT
  INTEGER TT,TX,TV,TH

```

```

COMPLEX K,C
COMPLEX U,V
COMPLEX Y

C
  DIMENSION K(PX,PT),C(PX,PF)
  DIMENSION U(PX,PT),V(PX,PF)
  DIMENSION A(PA),O(PA,PO)
  DIMENSION TT(PM),TX(PM),TV(PM),TH(PM)
  DIMENSION X(PW),Y(PW)

C
  COMMON/A10/CR,LP,FQ
  COMMON/A14/DCIC,DCIV
  COMMON/A15/DFTK,DFTU
  COMMON/B22/RLS,RCO,RTT,RCI,RTL,RFT
  COMMON/B23/WLS,WCO,WTT,WCI,WTL,WFT
  COMMON/D12/YI,YF,NY
  COMMON/D14/II,IF,NI
  COMMON/D15/ME,MX,MY,MW,MT
  COMMON/D18/IYI,IYF,NYI
  COMMON/D20/ITI,ITF,NTI

C
  10 FORMAT("1","1FT")

C
  IF(RFT.NE.0) CALL FTOR(DCIC,C,PX,PF)
  IF(RFT.NE.0) CALL FTOR(DCIV,V,PX,PF)

C
  WRITE(LP,10)
  CALL SRCFT(X,Y,PW,K,C,U,V,PX,PF,PT)

C
  CALL TPGK
  +(A,O,PA,PO,TT,TX,TV,TH,PM,K,PX,PT,
  + YI,YF,NY,MY,IYI,IYF,NYI,II,IF,NI,MT,ITI,ITF,NTI)
  CALL TPGK
  +(A,O,PA,PO,TT,TX,TV,TH,PM,U,PX,PT,
  + YI,YF,NY,MY,IYI,IYF,NYI,II,IF,NI,MT,ITI,ITF,NTI)

C
  IF(WFT.NE.0) CALL FTOW(DFTK,K,PX,PT)
  IF(WFT.NE.0) CALL FTOW(DFTU,U,PX,PT)

C
  RETURN
  END

C#####
C  SUBROUTINES
C#####
  SUBROUTINE SRC

C
C  CONSTANTS

C
  COMMON/F11/URE,ERE,UE,EE,OE
  COMMON/F12/URI,ERI,UI,EI,OI
  COMMON/F13/URA,ERA,UA,EA,OA
  COMMON/F14/UR2,ER2,U2,E2,O2
  COMMON/F15/URS,ERS,US,ES,OS
  COMMON/F16/UR1,ER1,U1,E1,O1
  COMMON/F17/URW,ERW,UW,EW,OW

C
  UE=UF(URE) $EE=EF(ERE)
  UI=UF(URI) $EI=EF(ERI)
  UA=UF(URA) $EA=EF(ERA)
  U2=UF(UR2) $E2=EF(ER2)
  US=UF(URS) $ES=EF(ERS)
  U1=UF(UR1) $E1=EF(ER1)
  UW=UF(URW) $EW=EF(ERW)

```

```

      RETURN
      END
C*****
      SUBROUTINE SRSC
C
C      CONSTANTS
C
      INTEGER CP,LP,PQ
C
      COMMON/A10/CP,LP,PQ
      COMMON/C15/APS,BPS,CPS
      COMMON/C16/ARX,BRX,CRX
      COMMON/C17/ARS,BRS,CRS
      COMMON/C20/PGPS,PGRX,PGFS
      COMMON/C21/PSPS,DCPS,DKPS,MPS,NPS
      COMMON/C22/PSRX,DCRX,DKRX,MRX,NRX
      COMMON/C23/PSRS,DCRS,DKRS,MRS,NRS
      COMMON/C24/TPPS,TRPS,TDPS,TPFS,TCFS
      COMMON/C25/TSRX,TRRX,TORX,TPRX,TCRX
      COMMON/C26/TSRS,TRRS,TDPS,TRFS,TCFS
C
      1 FORMAT(10I10)
      11 FORMAT("0","ALPHA          PRESTRIKE          ",1PG20.10/
+           " ","BETA          PRESTRIKE          ",1PG20.10/
+           " ","GAMMA         PRESTRIKE          ",1PG20.10/
+           "J","PULSE PEAK     PRESTRIKE          ",1PG20.10/
+           " ","PULSE MAXIMUM PRESTRIKE          ",1PG20.10/
+           "J","PULSE GUESS    PRESTRIKE          ",1PG20.10/
      12 FORMAT("0","ALPHA          RETURN           ",1PG20.10/
+           " ","BETA          RETURN           ",1PG20.10/
+           " ","GAMMA         RETURN           ",1PG20.10/
+           "J","PULSE PEAK     RETURN           ",1PG20.10/
+           " ","PULSE MAXIMUM RETURN           ",1PG20.10/
+           "J","PULSE GUESS    RETURN           ",1PG20.10/
      13 FORMAT("0","ALPHA          RESTRIKE          ",1PG20.10/
+           " ","BETA          RESTRIKE          ",1PG20.10/
+           " ","GAMMA         RESTRIKE          ",1PG20.10/
+           "J","PULSE PEAK     RESTRIKE          ",1PG20.10/
+           " ","PULSE MAXIMUM RESTRIKE          ",1PG20.10/
+           "J","PULSE GUESS    RESTRIKE          ",1PG20.10/
C
      IF(NPS.NE.0) CALL SRCS
+ (TRPS,TDPS,TPPS,TCPS,DCPS,DKPS,MPS,PSPS,PMPS,PGPS,APS,BPS,CPS)
      IF(NRX.NE.0) CALL SRCS
+ (TRRX,TORX,TPRX,TCRX,DCRX,DKRX,MRX,PSRX,PMRX,PGRX,ARX,BRX,CRX)
      IF(NRS.NE.0) CALL SRCS
+ (TRRS,TDRS,TRRS,TCRS,DCRS,DKRS,MRS,PSRS,PMRS,PGRS,ARS,BRS,CRS)
C
      READ(CP,1) MLC $IF(MLC.EQ.0) RETURN
C
      IF(NPS.NE.0) WRITE(LP,11) APS,BPS,CPS,PSPS,PMPS,PGPS
      IF(NRX.NE.0) WRITE(LP,12) ARX,BRX,CRX,PSRX,PMRX,PGRX
      IF(NRS.NE.0) WRITE(LP,13) ARS,BRS,CRS,PSRS,PMRS,PGRS
C
      RETURN
      END
C*****
      SUBROUTINE SRCS(TR,TD,TF,TC,DC,DK,ME,CO,CC,CG,A,B,C)
C
C      CONSTANTS
C
      DOUBLE PRECISION AT,EMAT
      DOUBLE PRECISION BT,EMBT

```

```

      REAL    NRF
C
      COMMON/C13/TSX,TRX,TDX,TPX,TCX
      COMMON/C14/AO,BO,XO
C
      EXTERNAL COF
      EXTERNAL CODF
C
      TRX=TR $TDX=TD $T=TD-TR
C
      AO=-ALOG(DC)/T    $A=AO
      BO=AO*CG          $B=NRF(BO,COF,CODF)
      C=0 $IF(ME,NE,0) C=-ALOG(DK)/TC
C
      AT=A*TR $EMAT=DEXP(-AT)
      BT=B*TR $EMBT=DEXP(-BT)
C
      CC=CO*(EMAT-EMBT)
C
      RETURN
      END
C*****
      SUBROUTINE SRLS(P,Q,PS)
C
C      PULSE
C
      INTEGER PS
      COMPLEX P,Q
C
      DIMENSION P(PS),Q(PS)
C
      CALL SRLK(P,PS)
      CALL SRLQ(Q,PS)
C
      RETURN
      END
C*****
      SUBROUTINE SRLK(K,PS)
C
C      PULSE
C
      INTEGER PS
      REAL    KF
      COMPLEX K
C
      DIMENSION K(PS)
C
      COMMON/C11/TM,WM
      COMMON/C12/NK,NC
      COMMON/C15/ME,MX,MY,MW,MS
C
      TI=0 $TF=TM $NT=NK $MT=MS
C
      DO 100 IT=1,NT,1
      T=AF(TI,TF,NT,IT,MT) $K(IT)=KF(T)
100  CONTINUE
C
      RETURN
      END
C*****
      SUBROUTINE SRLQ(C,PS)
C
C      PULSE
C

```

```

      INTEGER PS
      COMPLEX C,CF
C
      DIMENSION C(PS)
C
      COMMON/C11/T1,W1
      COMMON/C12/N1,K1,NC
      COMMON/D15/ME,IX,MY,MF,MT
C
      W1=0 $W1=W1 $N1=NC $MW=MF
C
      DO 100 IW=1,NW,1
      W=AF(W1,W1,N1,IX,MW) $C(IW)=CF(W)
100 CONTINUE
C
      RETURN
      END
C*****
      SUBROUTINE SRS(E,PC,ZC,PE,PF)
C
C      STROKE
C
      INTEGER PE,PF
      COMPLEX E
      COMPLEX PC,ZC
      COMPLEX DF
C
      DIMENSION E(PE,PF)
      DIMENSION PC(PF),ZC(PF)
C
      COMMON/D10/SI,SF,NS
      COMMON/D11/XI,XF,NX
      COMMON/D13/WI,W1,NW
      COMMON/D15/IS,IX,MY,MW,MT
      COMMON/D22/UI,UF,NU,MU
      COMMON/D23/VI,VF,NV,MV
      COMMON/D24/ZI,ZF,NZ,MZ
C
      UI=XI $VI=SI $ZI=WI
      UF=XF $VF=SF $ZF=W1
      NU=NX $NV=NS $NZ=NW
      MU=MX $MV=MS $MZ=MW
C
      DO 100 IW=1,NW,1
      W=AF(WI,W1,NW,IX,MW)
      CALL SRY(W,PC(IW),ZC(IW))
      DO 100 IS=1,NS,1
      S=AF(SI,SF,NS,IS,MS)
      E(IS,IW)=DF(S,W)
100 CONTINUE
C
      RETURN
      END
C*****
      SUBROUTINE SRY(W,PC,ZC)
C
C      PROPAGATION CONSTANT/CHARACTERISTIC IMPEDANCE
C
C      PARAMETER PR=10
C
      INTEGER PR
      INTEGER CR,LP,PQ
      COMPLEX WN,WNF
      COMPLEX W1,W1F

```

```

COMPLEX PKG,ZKG
COMPLEX RTS
COMPLEX TPC
COMPLEX ZX,ZL,ZT
COMPLEX YS,ZS
COMPLEX KI,KIF
COMPLEX KC,KCF
COMPLEX PC,ZC
COMPLEX DR
LOGICAL TEST

C
    DIMENSION RTS(10)

C
    COMMON/A10/CR,LP,PQ
    COMMON/B11/IMX,EMX
    COMMON/B27/MR
    COMMON/B30/MX
    COMMON/D21/XX,WX
    COMMON/E13/RAM,RAP
    COMMON/E14/KS
    COMMON/F11/URE,ERE,UE,EE,OE
    COMMON/H11/KC,KI

C
    EXTERNAL PKF

C
    DATA PR/10/
    DATA R2/1.41421356/
    DATA NPR/0/
    DATA NR /1/
    DATA TEST/.FALSE./
    DATA ES/1.0/

C
10  FORMAT("0","FREQUENCY",1PG20.10)
11  FORMAT("0","FREQUENCY",1PG20.10)
    + " ","WAVE NUMBER",2(1PG20.10)
    + " ","WAVE IMPEDANCE",2(1PG20.10)
    + " ","PROPAGATION CONSTANT",2(1PG20.10)
    + " ","PROPAGATION IMPEDANCE",2(1PG20.10)
    + " ","PROPAGATION CONSTANT",2(1PG20.10)
    + " ","ADMITTIVITY",2(1PG20.10)
    + " ","IMPEDIVITY",2(1PG20.10)
    + " ","CHARACTERISTIC CONSTANT",2(1PG20.10)
    + " ","CHARACTERISTIC IMPEDANCE",2(1PG20.10)
12  FORMAT(" ","PROPAGATION CONSTANT",2(1PG20.10))
13  FORMAT(" ","CHARACTERISTIC IMPEDANCE",2(1PG20.10))

C
    UC=UE REC=EE EOC=OE

C
    WN=WNF(W,UC,EO,OC) $FKG=WN/R2
    WI=WIF(W,UC,EO,OC) $ZKG=WI

C
    IF(MR.EQ.0) GO TO 100 $IF(W.EQ.0) GO TO 100

C
    WX=W
    RTS(1)=0
    CALL SRM(NPR,NR,RTS,IMX,EMX,EMX,PKF,TEST)
    TPC=RTS(1)
    CALL SRZ(W,TPC,ZX,ZL,ZT)
    YS=1/ZT
    ZS=ZX+ZL
    KC=KCF(YS,ZS)
    KI=KIF(YS,ZS)

C
    IF(MX.NE.0) WRITE(LP,11) W,WN,WI,FKG,ZKG,TPC,YS,ZS,KC,KI

```

```

C      PC=KC
      ZC=KI
C
      DR=PC-PKG
      ER=CABS(DR) $IF(CABS(PKG).NE.0) ER=CABS(DR/PKG)
      IF(ER.LT.E5) RETURN
C
100  CONTINUE
C
      WRITE(LP,10) W
      PC=PKG $WRITE(LP,12) PC
      ZC=ZKG $WRITE(LP,13) ZC
C
      RETURN
      END
C*****
      SUBROUTINE PKF(TPC,DET)
C
C      PROPAGATION CONSTANT
C
      INTEGER ORD
      COMPLEX CI
      COMPLEX EMW,EMI,EME
      COMPLEX WNW,WNI,WNE
      COMPLEX APCW,APCI,APCE
      COMPLEX WNF,APCF
      COMPLEX RIW,RIE
      COMPLEX TPC
      COMPLEX ARG
      COMPLEX BF,NE,HFM,HFP
      COMPLEX J0WW,J1WW
      COMPLEX J0IW,J1IW,Y0IW,Y1IW
      COMPLEX J0IS,J1IS,Y0IS,Y1IS
      COMPLEX H0ES,H1ES
      COMPLEX NJ,DJ,WJ
      COMPLEX NH,OH,WH
      COMPLEX DET
C
      COMMON/D21/X,W
      COMMON/E13/RX,RW
      COMMON/E14/RS
      COMMON/F11/JRE,ERE,UE,ER,OE
      COMMON/F12/URI,ERI,UI,EI,OI
      COMMON/F13/URW,ERW,UW,EW,OW
      COMMON/I10/WJ,WH
      COMMON/I11/J0WW,J1WW
      COMMON/I12/J0IW,J1IW,Y0IW,Y1IW
      COMMON/I13/J0IS,J1IS,Y0IS,Y1IS
      COMMON/I14/H0ES,H1ES
      COMMON/I15/EMW,EMI,EME
      COMMON/I16/APCW,APCI,APCE
C
      CI=CMPLX(0.0,1.0)
C
      EMW=OW+CI*W*EW
      EMI=OI+CI*W*EI
      EME=OE+CI*W*EE
C
      WNW=WNF(W,UW,EW,OW) $APCW=APCF(TPC,WNW)
      WNI=WNF(W,UI,EI,OI) $APCI=APCF(TPC,WNI)
      WNE=WNF(W,UE,EE,OE) $APCE=APCF(TPC,WNE)
C
      RIW=(EMW/EMI)*(APCI/APCW)

```

```

      RIE=(EME/EMI)*(APCI/APCE)
C
      ORD=0 $ARG=APCI*RW
      CALL BNHF(ORD,ARG,BF,NF,HFM,HFP)
      JUIW=BF $YUIW=NF
C
      ORD=1 $ARG=APCI*RW
      CALL BNHF(ORD,ARG,BF,NF,HFM,HFP)
      JLIW=BF $YLIW=NF
C
      ORD=0 $ARG=APCW*RW
      CALL BNHF(ORD,ARG,BF,NF,HFM,HFP)
      JOWW=BF
C
      ORD=1 $ARG=APCW*RW
      CALL BNHF(ORD,ARG,BF,NF,HFM,HFP)
      JIWW=BF
C
      ORD=0 $ARG=APCI*RS
      CALL BNHF(ORD,ARG,BF,NF,HFM,HFP)
      JUIS=BF $YUIS=NF
C
      ORD=1 $ARG=APCI*RS
      CALL BNHF(ORD,ARG,BF,NF,HFM,HFP)
      JIIS=BF $YIIS=NF
C
      ORD=0 $ARG=APCE*RS
      CALL BNHF(ORD,ARG,BF,NF,HFM,HFP)
      HOES=HFP
C
      ORD=1 $ARG=APCE*RS
      CALL BNHF(ORD,ARG,BF,NF,HFM,HFP)
      HIES=HFP
C
      NJ=RIW*JIWW*JUIW-JOWW*JLIW $NH=RIE*HIES*JUIS-HOES*JIIS
      DJ=RIW*JIWW*YUIW-JOWW*YLIW $DH=RIE*HIES*YUIS-HOES*YIIS
      WJ=NJ/DJ $WH=NH/DH
C
      DET=WJ-WH
C
      RETURN
      END
C*****
      SUBROUTINE SRZ(W,TPC,ZS,ZL,ZT)
C
C      IMPEDANCES
C
      COMPLEX CI
      COMPLEX TPC
      COMPLEX EMW,EMI,EME
      COMPLEX APCW,APCI,APCE
      COMPLEX JOWW,JIWW
      COMPLEX JUIW,JLIW,YUIW,YLIW
      COMPLEX JUIS,JIIS,YUIS,YIIS
      COMPLEX HUES,HIES
      COMPLEX WJ,WH
      COMPLEX PC
      COMPLEX DJ,DY,DH
      COMPLEX DP,DS,DW
      COMPLEX RP,RQ,RL
      COMPLEX CC
      COMPLEX CS,CLP,CLQ,CT
      COMPLEX CJY,CH
      COMPLEX ZS

```

```

COMPLEX ZLP,ZLQ,ZL
COMPLEX ZTP,ZTQ,ZT

C
COMMON/I13/RX,RW
COMMON/F11/URE,EFE,UE,EL,OE
COMMON/F12/URL,EKI,UI,EL,OI
COMMON/I10/WJ,WH
COMMON/I11/JJWW,J1WW
COMMON/I12/JJ1W,J11W,YJ1W,Y11W
COMMON/I13/JJ1S,J11S,YJ1S,Y11S
COMMON/I14/HUES,H1ES
COMMON/I15/EMW,EMI,EME
COMMON/I16/APCW,APCI,APCE

C
DATA PI/3.14159265/

C
CI=CMPLX(0.,1.0)

C
PC=WJ

C
DJ=(JJ1S-JJ1W)/J1WW
DY=(YJ1S-YJ1W)/J1WW
DH=HUES/J1WW

C
DR=JJWW/J1WW
DS=JJ1S-PC*YJ1S
DW=J11W-PC*Y11W

C
RP=(EMW/EMI)*(APCW/APCI)*(J1WW/DW)
RQ=-(EMW/EMI)*(APCW/APCI)*(J1WW/DW)*PC
RE=(EMW/EMI)*(APCI/APCE)*(APCW/APCE)*(J1WW/HUES)*(DS/DW)

C
CC=2*PI*RW*EMW*APCW

C
CS=(APCW*APCW)/CC
CLP=(CI*W*UI*EMI)/CC
CLQ=(CI*W*UE*EME)/CC
CT=1/CC

C
CJY=RP*DJ+RQ*DY
CH=RE*DH

C
ZLP=-CLP*CJY  $ZTP=-CT*CJY  $ZS=CS*DR
ZLQ=CLQ*CH  $ZTQ=CT*CH
ZL=ZLP+ZLQ  $ZT=ZTP+ZTQ

C
RETURN
END
C*****
SUBROUTINE SRCC(E,PC,ZC,IT,ME,PE,PF)

C
C
C COAXIAL CABLE

C
INTEGER PE,PF
INTEGER CR,LP,PQ
COMPLEX WNX,WNF
COMPLEX WIX,WIF
COMPLEX E,IT
COMPLEX PC,ZC
COMPLEX MR

C
DIMENSION E(PE,PF),IT(PE,PF)
DIMENSION PC(PF),ZC(PF)
DIMENSION MR(PF)

```

```

C
COMMON/A10/CR,LP,PQ
COMMON/B30/MC
COMMON/D11/XI,XF,NX
COMMON/D12/YI,YF,NY
COMMON/D13/WI,WF,NW
COMMON/D15/MS,MX,MY,MW,MT
COMMON/D22/UI,UF,NU,MU
COMMON/D23/VI,VF,NV,MV
COMMON/D24/ZI,ZF,NZ,MZ
COMMON/F16/UR,ER,U1,E1,O1

C
10 FORMAT("0","FREQUENCY",1PG20.10)
12 FORMAT(" ","PROPAGATION CONSTANT",2(1PG20.10))
13 FORMAT(" ","CHARACTERISTIC IMPEDANCE",2(1PG20.10))

C
UC=U1 $EC=E1 $OC=O1

C
UI=YI $VI=XI $ZI=WI
UF=YF $VF=XF $ZF=WF
NU=NY $NV=NX $NZ=NW
MU=MY $MV=MX $MZ=MW

C
DO 101 IW=1,NW,1
W=AF(WI,WF,NW,IW,MW) $IF(MC.NE.0) WRITE(LP,10) W
WNX=WNF(W,UC,EC,OC) $PC(IW)=WNX $IF(MC.NE.0) WRITE(LP,12) WNX
WIX=WIF(W,UC,EC,OC) $ZC(IW)=WIX $IF(MC.NE.0) WRITE(LP,13) WIX
DO 102 IX=1,NX,1
E(IX,IW)=IT(IX,IW)*MR(IW)
102 CONTINUE
101 CONTINUE

C
RETURN
END

C*****
SUBROUTINE SRLT(X,Y,PW,C,V,E,PC,ZC,PE,PX,PF)

C
C TRANSMISSION LINE
C
INTEGER PE,PX,PF
INTEGER PW
COMPLEX C,V
COMPLEX E
COMPLEX PC,ZC
COMPLEX WNX,WIX
COMPLEX ZMX,ZPX
COMPLEX ZLM,ZLP
COMPLEX CKF,VKF
COMPLEX Y
COMPLEX QF
LOGICAL TEST

C
DIMENSION C(PX,PF),V(PX,PF)
DIMENSION E(PE,PF)
DIMENSION PC(PF),ZC(PF)
DIMENSION X(PW),Y(PW)

C
COMMON/B15/NQ,NI
COMMON/B31/MM
COMMON/D21/XX,ZX
COMMON/D22/XI,XF,NX,MX
COMMON/D23/YI,YF,NY,MY
COMMON/D24/ZI,ZF,NZ,MZ
COMMON/G11/7LM,ZLP

```

```

COMMON/H13/WNX,WIX
COMMON/H14/ZMX,ZPX
C
EXTERNAL CKF,VKF
C
DATA IM/0/
C
CALL SRAF(X,PW,YI,YF,NY,MY)
C
DO 100 IF=1,NZ,1
ZX=AF(ZI,ZF,NZ,IP,MZ)
WNX=PC(IP)
WIX=ZC(IP)
ZMX=ZLM $IF(MM.NE.0) ZMX=WIX
ZPX=ZLP $IF(MM.NE.0) ZPX=WIX
CALL SRD(IM,IP,E,PE,PF,Y,PW)
DO 100 IX=1,NX,1
XX=AF(XI,XF,NX,IX,MX)
TEST=(MM.EQ.0).OR.(MM.EQ.1)
IF(TEST) C(IX,IP)=QF(YI,YF,NY,MY,NQ,CKF,X,Y,PW)
TEST=(MM.EQ.0).OR.(MM.EQ.2)
IF(TEST) V(IX,IP)=QF(YI,YF,NY,MY,NQ,VKF,X,Y,PW)
100 CONTINUE
C
RETURN
END
C*****
SUBROUTINE SRT(MR,PF)
C
C TRANSFER
C
INTEGER PF
INTEGER CR,LP,PQ
REAL MSI,MTA,MTD,MTS,MIT
COMPLEX SI,SIF
COMPLEX TA,TD,TS,TF
COMPLEX TT
COMPLEX MR
C
DIMENSION MR(PF)
C
COMMON/A10/CR,LP,PQ
COMMON/D13/WI,WF,NW
COMMON/D15/MS,MX,NY,MW,MT
COMMON/E12/RSM,RSP
COMMON/E13/RAM,RAP
COMMON/F13/URA,ERA,UA,EA,OA
COMMON/F14/UR2,ER2,U2,E2,O2
COMMON/F15/URS,ERS,US,ES,OS
C
10 FORMAT("0"," INDEX",1X," FREQUENCY",1X," SURFACE",1X,
+ " ARMOR",1X," SPACER",1X," SHIELD",1X,
+ " TOTAL" //)
11 FORMAT(" ",I10,1X,6(1PG10.3,1X))
C
WRITE(LP,10)
DO 100 IW=1,NW,1
W=AF(WI,WF,NW,IW,MW) $IF(W.EQ.0) GO TO 100
SI=SIF(RAP,W,UA,EA,OA) $MSI=CABS(SI)
TA=TF(RAM,RAP,W,UA,EA,OA) $MTA=CABS(TA)
TD=TF(RSP,RAM,W,U2,E2,O2) $MTD=CABS(TD)
TS=TF(RSM,RSP,W,US,ES,OS) $MTS=CABS(TS)
TT=SI*TA*TD*TS $MR(IW)=TT $MTT=CABS(TT)
WRITE(LP,11) IW,W,MSI,MTA,MTD,MTS,MTT

```

```

100 CONTINUE
C
    RETURN
    END
C*****
    SUBROUTINE SRD(IX,IY,W,PX,PY,Z,PZ)
C
C    DATA
C
    INTEGER PX,PY
    INTEGER PZ
    COMPLEX W,Z
C
    DIMENSION W(PX,PY)
    DIMENSION Z(PZ)
C
    IF(IX.EQ.0) GO TO 101
    IF(IY.EQ.0) GO TO 102
C
101 CONTINUE
    DO 110 IS=1,PX,1
        Z(IS)=W(IS,IY)
110 CONTINUE
    RETURN
C
102 CONTINUE
    DO 120 IS=1,PY,1
        Z(IS)=W(IX,IS)
120 CONTINUE
    RETURN
C
    END
C*****
    SUBROUTINE SRCFT(X,Y,PW,K,C,U,V,PX,PF,PT)
C
C    FOURIER TRANSFORM (CONTINUOUS)
C
    INTEGER PX,PF,PT
    INTEGER PW
    COMPLEX K,C
    COMPLEX U,V
    COMPLEX IKF
    COMPLEX Y
    COMPLEX IF
C
    DIMENSION K(PX,PT),C(PX,PF)
    DIMENSION U(PX,PT),V(PX,PF)
    DIMENSION X(PW),Y(PW)
C
    COMMON/B15/NQ,NI
    COMMON/D12/YI,YF,NY
    COMMON/D13/WI,WF,NW
    COMMON/D14/VI,VF,NT
    COMMON/D15/MS,MX,MY,MW,MT
    COMMON/D21/IX,XX
C
    EXTERNAL IKF
C
    DATA IM/0/
C
    CALL SRAF(X,PW,WI,WF,NW,MW)
C
    DO 100 IY=1,NY,1
        XX=AF(YI,YF,NY,IY,MY)

```

```

      CALL SRD(IY,IM,C,FX,PF,Y,PW)
      DO 101 IT=1,NT,1
      TX=AF(TI,TF,NT,IT,MT)
      <(IY,IT)=IF(WI,WF,NW,MW,NI,IKF,X,Y,PW)
101  CONTINUE
      CALL SRD(IY,IM,V,FX,PF,Y,PW)
      DO 102 IT=1,NT,1
      TX=AF(TI,TF,NT,IT,MT)
      U(IY,IT)=IF(WI,WF,NW,MW,NI,IKF,X,Y,PW)
102  CONTINUE
100  CONTINUE
C
      RETURN
      END
C*****
      SUBROUTINE GRAF(A,PF,XI,XF,NX,MX)
C
C      ARRAY
C      ?
C      INTEGER PF
C
C      DIMENSION A(PF)
C
C      DO 100 IX=1,NX,1
      XO=AF(XI,XF,NX,IX,MX) $A(IX)=XO
100  CONTINUE
C
      RETURN
      END
C#####
C      FUNCTIONS
C#####
      REAL FUNCTION UF(UR)
C
C      PERMEABILITY
C
C      DATA UO/12.5664E-07/
C
C      U=UO*UR $UF=U
C
      RETURN
      END
C*****
      REAL FUNCTION EF(ER)
C
C      PERMITTIVITY
C
C      DATA EO/8.8542E-12/
C
C      E=EO*ER $EF=E
C
      RETURN
      END
C*****
      REAL FUNCTION KF(T)
C
C      CURRENT
C
C      INTEGER PS
C
      COMMON/C15/APS,BPS,CPS
      COMMON/C16/ARX,BRX,CRX
      COMMON/C17/ARS,BRS,CRS
      COMMON/C21/PSPS,DCPS,DKPS,MPS,NPS

```



```

      TR=TS+IS*TP  BK=KS(PS,TR,T,A,B)
      TT=TR-TS  BAK=SGN*C*TT  BTERM=K*DEXP(AX)
      SUM=SUM+TERM
100  CONTINUE
      STF=SUM
C
      RETURN
      END
C*****
      COMPLEX FUNCTION SFF(PS,TS,TP,W,A,B,C,SGN,NR)
C
C      STROKE (DOUBLE EXPONENTIAL)
C
      INTEGER SGN
      COMPLEX CI
      COMPLEX CA,CB,CC
      COMPLEX AS,AT
      COMPLEX CS,CT
      COMPLEX TERM,SUM
C
      CI=CMPLX(0.0,1.0)
C
      SFF=0  BIF(NR.EQ.0) RETURN
C
      CA=A+CI*W
      CB=B+CI*W
      CC=-SGN*C+CI*W
C
      CS=PS*(1/CA-1/CB)
      AS=-CI*W*TS  BCT=DEXP(AS)
C
      SUM=0
      DO 100 IR=1,NR,1  BIS=IR-1
      AT=-IS*CC*TP  BTERM=DEXP(AT)
      SUM=SUM+TERM
100  CONTINUE
      SFF=CS*CI*SUM
C
      RETURN
      END
C*****
      REAL FUNCTION KS(PS,TS,T,A,B)
C
C      CURRENT
C
      DOUBLE PRECISION AT,EMAT
      DOUBLE PRECISION BT,EMBT
C
      REAL K
C
      KS=0  BIF(T.LT.TS) RETURN
C
      TT=T-TS
      AT=A*TT  BEMAT=DEXP(-AT)
      BT=B*TT  BEMBT=DEXP(-BT)
      K=PS*(EMAT-EMBT)  BKS=K
C
      RETURN
      END
C*****
      REAL FUNCTION COF(X)
C
C      CURRENT

```

```

      DOUBLE PRECISION AT,EMAT
      DOUBLE PRECISION BT,EMBT
C
      COMMON/C13/TS,TR,TD,TP,TC
      COMMON/C14/A,B,C
C
      AT=A*TR $EMAT=DEXP(-AT)
      BT=X*TR $EMBT=DEXP(-BT)
C
      F=BT*EMBT-AT*EMAT $COF=F
C
      RETURN
      END
C*****
      REAL FUNCTION COF(X)
C
      CURRENT DERIVATIVE
C
      DOUBLE PRECISION AT,EMAT
      DOUBLE PRECISION BT,EMBT
C
      COMMON/C13/TS,TR,TD,TP,TC
      COMMON/C14/A,B,C
C
      AT=A*TR $EMAT=DEXP(-AT)
      BT=X*TR $EMBT=DEXP(-BT)
C
      FD=TR*(1-BT)*EMBT $COF=FD
C
      RETURN
      END
C*****
      COMPLEX FUNCTION WNF(W,U,E,O)
C
      WAVE NUMBER
C
      COMPLEX CI
      COMPLEX UC,EC
      COMPLEX WN2,WN
C
      CI=CMPLX(0.0,1.0)
C
      UC=U
      EC=E $IF(W.NE.0) EC=E-CI*(O/W)
C
      WN2=-W*W*UC*EC $WN=CSORT(WN2) $WNF=WN
C
      RETURN
      END
C*****
      COMPLEX FUNCTION WIF(W,U,E,O)
C
      WAVE IMPEDANCE
C
      COMPLEX CI
      COMPLEX UC,EC
      COMPLEX WI2,WI
C
      CI=CMPLX(0.0,1.0)
C
      UC=U
      EC=E $IF(W.NE.0) EC=E-CI*(O/W)
C
      WI2=+UC/EC $WI=CSORT(WI2) $WIF=WI

```

```

C      RETURN
C      END
C*****
C      COMPLEX FUNCTION KIF(Y,Z)
C      CHARACTERISTIC IMPEDANCE
C      COMPLEX Y,Z
C      COMPLEX KI2,KI
C      KI2=Z/Y $KI=CSQRT(KI2) $KIF=KI
C      RETURN
C      END
C*****
C      COMPLEX FUNCTION KCF(Y,Z)
C      PROPAGATION CONSTANT
C      COMPLEX Y,Z
C      COMPLEX KC2,KC
C      KC2=Y*Z $KC=CSQRT(KC2) $KCF=KC
C      RETURN
C      END
C*****
C      COMPLEX FUNCTION APCF(TPC,WN)
C      AXIAL PROPAGATION CONSTANT
C      COMPLEX TPC,WN
C      COMPLEX APC
C      APC=CSQRT(TPC*TPC-WN*WN) $APCF=APC
C      RETURN
C      END
C*****
C      COMPLEX FUNCTION DF(X,W)
C      DIPOLE
C      REAL KO
C      COMPLEX CO,CF
C      COMPLEX EO
C      COMPLEX E
C      COMMON/E15/ZD,YT
C      COMMON/F11/URE,ERE,UE,EE,OE
C      DATA PI/3.14159265/
C      UC=UE $EC=EE $OC=OE
C      KO=1/(2*PI*UC) $CO=CF(W) $EO=KO*CO
C      R2=X*X+YT*YT+ZD*ZD $R=SQRT(R2)
C      E=EO*(X/(R**3)) $DF=E
C      RETURN
C      END
C***** A-28 *****
C      COMPLEX FUNCTION SIF(R,W,U,E,O)

```

```

C
C SURFACE IMPEDANCE
C
C INTEGER ORD
C COMPLEX WN,WNF
C COMPLEX WI,WIF
C COMPLEX ARG
C COMPLEX MBF,MNF,MHEM,MHEP
C COMPLEX IO,I1
C COMPLEX DR
C COMPLEX SI
C
C DATA PI/3.14159265/
C
C WN=WNF(W,U,E,0)
C WI=WIF(W,U,E,0)
C
C ORD=0 $ARG=WN*R
C CALL MBNHF(ORD,ARG,MBF,MNF,MHEM,MHEP)
C IO=MBF
C
C ORD=1 $ARG=WN*R
C CALL MBNHF(ORD,ARG,MBF,MNF,MHEM,MHEP)
C I1=MBF
C
C DR=IO/I1 $SI=(WI/(2*PI*F))*DR $SIF=SI
C
C RETURN
C END
C *****
C COMPLEX FUNCTION TF(RI,RO,W,U,E,0)
C
C TRANSFER RATIOS
C
C INTEGER ORD
C COMPLEX WN,WNF
C COMPLEX ARG
C COMPLEX MBF,MNF,MHEM,MHEP
C COMPLEX IO,K00
C COMPLEX IO1,K01
C COMPLEX IO,K10
C COMPLEX IO1,K11
C COMPLEX DET,WIR,WJR
C COMPLEX IR,JR
C
C DATA PI/3.14159265/
C
C WN=WNF(W,U,E,0)
C
C IO0=0 $K00=0
C IO1=0 $K01=0
C IO0=0 $K10=0
C IO1=0 $K11=0
C
C ORD=0 $ARG=WN*RO
C CALL MBNHF(ORD,ARG,MBF,MNF,MHEM,MHEP)
C IO0=MBF $K00=MHEM
C
C ORD=1 $ARG=WN*RO
C CALL MBNHF(ORD,ARG,MBF,MNF,MHEM,MHEP)
C IO0=MBF $K10=MHEM
C
C ORD=0 $ARG=WN*RI
C CALL MBNHF(ORD,ARG,MBF,MNF,MHEM,MHEP)

```

```

      IOI=MBF *K0I=MHFM
C
      ORD=1 $A2G=WN*RI
      CALL M3MF(ORD,AFG,MBF,MNF,MHFM,MHFP)
      I1I=MBF $K1I=MHFM
C
      DET=I00 $IF(RI.NE.0) DET=I00*K1I+I1I*K00
      WIR=I10 $IF(RI.NE.0) WIR=I10*K1I-I1I*K10
      WJR=1 $IF(RI.NE.0) WJR=I0I*K1I+I1I*K0I
C
      IR=((2*PI*RO)/WN)*(WIR/DET)
      JR= WJR/DET $TF=JR
C
      RETURN
      END
C*****
      COMPLEX FUNCTION CKF(XS,XP,X,Y,PW)
C
C      KERNAL (CURRENT)
C
      INTEGER PW
      COMPLEX Y
      COMPLEX EX,EDX
      COMPLEX GF,GGF
      COMPLEX KF
C
      DIMENSION X(PW),Y(PW)
C
      COMMON/B14/ND,MD,MC
      COMMON/D23/XI,XF,NX,MX
C
      CALL TLU(X,Y,PW,NX,ND,XP,EX,EDX,MD,MC)
      GF=GGF(XS,XP) $KF=EX*GF $CKF=KF
C
      RETURN
      END
C*****
      COMPLEX FUNCTION VKF(XS,XP,X,Y,PW)
C
C      KERNAL (VOLTAGE)
C
      INTEGER PW
      COMPLEX Y
      COMPLEX EX,EDX
      COMPLEX GF,VGF
      COMPLEX KF
C
      DIMENSION X(PW),Y(PW)
C
      COMMON/B14/ND,MD,MC
      COMMON/D23/XI,XF,NX,MX
C
      CALL TLU(X,Y,PW,XI,ND,XP,EX,EDX,MD,MC)
      GF=VGF(XS,XP) $KF=EX*GF $VKF=KF
C
      RETURN
      END
C*****
      COMPLEX FUNCTION CGF(X,XP)
C
C      GREENS FUNCTION (CURRENT)
C
      INTEGER CP,LP,PQ
      REAL L

```

```

COMPLEX AML,AYM,AYP,AZM,AZP
COMPLEX EML,EYM,EYP,EZM,EZP
COMPLEX DC,DET
COMPLEX PC,ZC
COMPLEX ZLM,ZLP
COMPLEX RCM,RCP,RCF
COMPLEX DCEXP
COMPLEX GF
LOGICAL LGT,LLT,LTT

C
COMMON/A10/CR,LP,PQ
COMMON/D22/XI,XF,NX,MX
COMMON/D23/YI,YF,NY,MY
COMMON/H13/PC,ZC
COMMON/H14/ZLM,ZLP

C
10 FORMAT("0","ERROR (CGF) : PARAMETER LIMITS NOT SATISFIED",
+ " ",10(1PG20.10)/)

C
LGT=(X.GT.XF).OR.(XP.GT.YF)
LLT=(X.LT.XI).OR.(XP.LT.YI)
LTT=LGT.OR.LLT
IF(LTT) WRITE(LP,10) XI,X,XF,YI,XP,YF $IF(LTT) STOP

C
L=YF-YI $SI=0 $SF=L

C
Y=SXY(YI,YF,SI,SF,X)
YP=SXY(YI,YF,SI,SF,XP)

C
AD=Y-YP $YD=ABS(AD) $ZD=2*L-YD
AS=Y+YP $YS=ABS(AS) $ZS=2*L-YS

C
AML=-2*PC*L $EML=DCEXP(AML)

C
AYM=-PC*YD $EYM=DCEXP(AYM)
AYP=-PC*YS $EYP=DCEXP(AYP)
AZM=-PC*ZD $EZM=DCEXP(AZM)
AZP=-PC*ZS $EZP=DCEXP(AZP)

C
RCM=RCF(ZLM,ZC)
RCP=RCF(ZLP,ZC)

C
IF(X.LT.XP) DC=EYM+RCM*RCP*EZM-RCM*EYP-RCP*EZP
IF(X.GT.XP) DC=EYM+RCM*RCP*EZM-RCM*EYP-RCP*EZP

C
DET=2*(1-RCM*RCP*EML)

C
GF=(DC/DET)/ZC $CGF=GF

C
RETURN
END
C*****
COMPLEX FUNCTION VGF(X,XP)
C
C GREENS FUNCTION (VOLTAGE)
C
INTEGER CR,LP,PQ
REAL L
COMPLEX AML,AYM,AYP,AZM,AZP
COMPLEX EML,EYM,EYP,EZM,EZP
COMPLEX DV,DET
COMPLEX PC,ZC
COMPLEX ZLM,ZLP
COMPLEX RCM,RCP,RCF

```

```

      COMPLEX DCEXP
      COMPLEX GF
      LOGICAL LGT,LLT,LTT
C
      COMMON/A10/CR,LP,PQ
      COMMON/D22/XI,XF,NX,MX
      COMMON/D23/YI,YF,NY,MY
      COMMON/H13/PC,ZC
      COMMON/H14/ZLM,ZLP
C
      10 FORMAT("U","ERROR (VGF) : PARAMETER LIMITS NOT SATISFIED",
+           " ",10(1PG20.10)/)
C
      LGT=(X.GT.XF).OR.(XP.GT.YF)
      LLT=(X.LT.XI).OR.(XP.LT.YI)
      LTT=LGT.OR.LLT
      IF(LTT) WRITE(LP,10) XI,X,XF,YI,YF $IF(LTT) STOP
C
      L=YF-YI $SI=0 $SF=L
C
      Y=SKY(YI,YF,SI,SF,X)
      YP=SKY(YI,YF,SI,SF,XP)
C
      AD=Y-YP $YD=ABS(AD) $ZD=2*L-YD
      AS=Y+YP $YS=ABS(AS) $ZS=2*L-YS
C
      AML=-2*PC*L $EML=DCEXP(AML)
C
      AYM=-PC*YD $EYM=DCEXP(AYM)
      AYP=-PC*YS $EYP=DCEXP(AYP)
      AZM=-PC*ZD $EZM=DCEXP(AZM)
      AZP=-PC*ZS $EZP=DCEXP(AZP)
C
      RCM=RCF(ZLM,ZC)
      RCP=RCF(ZLP,ZC)
C
      IF(X.LT.XP) DV=EYM-RCM*RCF*EZM+RCM*EYP-RCP*EZP
      IF(X.GT.XF) DV=EYM-RCM*RCF*EZM-RCM*EYP+RCP*EZP
C
      DET=2*(1-RCM*RCP*EML)
C
      GF=DV/DET $VGF=GF
C
      RETURN
      END
C*****
      REAL FUNCTION SKY(XI,XF,YI,YF,X)
C
      SCALING
C
      S=(YF-YI)/(XF-XI) $Y=S*(X-XI)+YI $SKY=Y
C
      RETURN
      END
C*****
      COMPLEX FUNCTION RCF(ZL,ZC)
C
      REFLECTION COEFFICIENT
C
      COMPLEX ZL,ZC
      COMPLEX RC
C
      RC=(ZL-ZC)/(ZL+ZC) $RCF=RC

```

```

      RETURN
      END
C*****
      COMPLEX FUNCTION IKF(XD,XI,X,Y,PW)
C
C      INVERSE KERNAL FUNCTION
C
      INTEGER PW
      REAL    KR,KI
      COMPLEX Y
      COMPLEX KX,KDX
      COMPLEX GF,IGF
      COMPLEX KF
C
      DIMENSION X(PW),Y(PW)
C
      COMMON/B14/ND,MD,MX
      COMMON/D13/WI,WF,NW
C
      DATA PI/3.14159265/
C
      CALL TLU(X,Y,PW,NW,ND,XI,KX,KDX,MD,MX) $KR=REAL(KX) $KI=AIMAG(KX)
      GF=IGF(XD,XI) $GR=REAL(GF) $GI=AIMAG(GF)
C
      KF=2*(KR*GR-KI*GI)/(2*PI) $IKF=KF
C
      RETURN
      END
C*****
      COMPLEX FUNCTION IGF(X,XT)
C
C      INVERSE GREENS FUNCTION
C
      INTEGER SGN
      COMPLEX CI
      COMPLEX CA
      COMPLEX DCEXP
      COMPLEX GF
C
      DATA SGN/+1/
C
      CI=CMPLX(0.0,1.0)
C
      CA=SGN*CI*X*XT $GF=DCEXP(CA) $IGF=GF
C
      RETURN
      END
C#####
C      LIBRARY
C#####
      REAL FUNCTION AF(XI,XF,NX,IX,MX)
C
C      ARGUMENTS
C
      INTEGER CR,LP,PQ
      REAL    LSF,LNF,LTF
C
      COMMON/A16/CR,LP,PQ
C
      10 FORMAT("0","ERROR (AF) : PARAMETER LIMITS NOT SATISFIED"//)
C
      IF(IX.LT.1 ) WRITE(LP,10)
      IF(IX.GT.NX) WRITE(LP,10)
C

```

```

      AF=XI $IF(NX.EQ.1) RETURN
C
      IF(MX.LT.0) X=LSF(XI,XF,NX,IX)
      IF(MX.EQ.0) X=LNF(XI,XF,NX,IX)
      IF(MX.GT.0) X=LTF(XI,XF,NX,IX)
C
      AF=X
C
      RETURN
      END
C*****
      REAL FUNCTION LSF(XI,XF,NX,IX)
C
C      ARGUMENTS
C
      INTEGER CR,LP,PQ
      REAL      N,D
C
      COMMON/A10/CR,LP,PQ
C
10  FORMAT("0","ERROR (LSF) : PARAMETER LIMITS NOT SATISFIED"/)
C
      IF(IX.LT.1 ) WRITE(LP,10)
      IF(IX.GT.NX) WRITE(LP,10)
C
      N=IX-1
      D=NX-1
      DIF=XF-XI
      X=XI+(N/D)*DIF $LSF=X
C
      RETURN
      END
C*****
      REAL FUNCTION LNF(XI,XF,NX,IX)
C
C      ARGUMENTS
C
      INTEGER CR,LP,PQ
      REAL      N,D
C
      COMMON/A10/CR,LP,PQ
C
      DATA BASE/2.7182818/
C
10  FORMAT("0","ERROR (LNF) : PARAMETER LIMITS NOT SATISFIED"/)
C
      IF(IX.LT.1 ) WRITE(LP,10)
      IF(IX.GT.NX) WRITE(LP,10)
C
      LNF=0 $IF((IX.EQ.1).AND.(XI.EQ.0)) RETURN
C
      XLI=0 $IF(XI.NE.0) XLI=ALOG(XI)
      XLF=0 $IF(XF.NE.0) XLF=ALOG(XF)
C
      N=IX-1
      D=NX-1
      DLX=XLF-XLI
      XL=XLI+(N/D)*DLX
      X=ALF(XL,BASE) $LNF=X
C
      RETURN
      END
C***** A-34 *****
      REAL FUNCTION LTF(XI,XF,NX,IX)

```

```

C
C ARGUMENTS
C
C INTEGER CR,LP,PQ
C REAL N,D
C
C COMMON/A10/CR,LP,PQ
C
C DATA BASE/10.0/
C
10 FORMAT("0","ERROR (LTF) : PARAMETER LIMITS NOT SATISFIED"/)
C
C IF(IX.LT.1) WRITE(LP,10)
C IF(IX.GT.NX) WRITE(LP,10)
C
C LTF=0 $IF((IX.EQ.1).AND.(XI.EQ.0)) RETURN
C
C XLI=0 $IF(XI.NE.0) XLI=ALOG10(XI)
C XLF=0 $IF(XF.NE.0) XLF=ALOG10(XF)
C
C N=IX-1
C D=NX-1
C DLX=XLF-XLI
C XL=XLI+(N/D)*DLX
C X=ALF(XL,BASE) $LTF=X
C
C RETURN
C END
C*****
C REAL FUNCTION ALF(ARG,BASE)
C
C ANTI-LOG
C
C AL=BASE**ARG $ALF=AL
C
C RETURN
C END
C*****
C SUBROUTINE SDB(D,PA,PO,NA,NO,ML,RL,SL)
C
C SCALE (DB)
C
C DOUBLE PRECISION A
C
C INTEGER PA,PO
C INTEGER CR,LP,PQ
C
C DIMENSION O(PA,PO)
C
C COMMON/A10/CR,LP,PQ
C
10 FORMAT("0","ERROR (SDB) : PARAMETER LIMITS NOT SATISFIED"/)
C
C IF(NA.GT.PA) WRITE(LP,10)
C IF(NO.GT.PO) WRITE(LP,10)
C
C IF(ML.EQ.0) RETURN
C
C DO 100 IA=1,NA,1
C DO 100 IO=1,NO,1
C AR=O(IA,IO)/RL $A=ABS(AR)
C O(IA,IO)=0 $IF(A.EQ.0) GO TO 100
C IF(ML.LT.0) O(IA,IO)=SL*DLOG (A)
C IF(ML.GT.0) O(IA,IO)=SL*DLOG10(A)

```

```

100 CONTINUE
C
  RETURN
  END
C*****
SUBROUTINE MNMX(O,PA,PO,NA,NO,XMN,XXM,MIN,MAX,MD)
C
C   MIN/MAX
C
  INTEGER PA,PO
  INTEGER CP,LP,PQ
  REAL    MIN,MAX
C
  DIMENSION O(PA,PO)
C
  COMMON/A10/CR,LP,PQ
C
10  FORMAT("U","ERROR (MNMX) : PARAMETER LIMITS NOT SATISFIED"//)
C
  IF(NA.GT.PA) WRITE(LP,10)
  IF(NO.GT.PO) WRITE(LP,10)
C
  AMN=O(1,1)
  AMX=O(1,1)
  DO 100 IA=1,NA,1
  DO 100 IO=1,NO,1
  AMN=AMIN1(AMN,O(IA,IO))
  AMX=AMAX1(AMX,O(IA,IO))
100 CONTINUE
C
  IF(MD) 101,102,103
101 CONTINUE
  XMN=AMN
  XXM=AMX
  GO TO 110
102 CONTINUE
  XMN=MIN
  XXM=MAX
  GO TO 110
103 CONTINUE
  XMN=MIN $IF(XMN.GT.AMN) XMN=AMN
  XXM=MAX $IF(XXM.LT.AMX) XXM=AMX
110 CONTINUE
  IF(XMN.GT.O) XMN=O
  IF(XXM.LT.O) XXM=O
C
  RETURN
  END
C*****
SUBROUTINE TBL(A,C,PA,PO,NA,NO)
C
C   TABLE
C
  INTEGER PA,PO
  INTEGER CP,LP,PQ
C
  DIMENSION A(PA),O(PA,PO)
C
  COMMON/A10/CR,LP,PQ
C
10  FORMAT("U","ERROR (TBL) : PARAMETER LIMITS NOT SATISFIED"//)
11  FORMAT(" ",I10,10(1PG20.10))
C
  IF(NA.GT.PA) WRITE(LP,10)

```

```

      IF(NO.GT.PO) WRITE(LP,10)
C
      DO 100 IA=1,NA,1
      WRITE(LP,11) IA,A(IA),(O(IA,IO),IO=1,NO,1)
100  CONTINUE
C
      RETURN
      END
C*****
      SUBROUTINE PLT(A,O,PA,PO,NA,NO,MIN,MAX)
C
C      PLOT
C
C      PARAMETER NS=9
C      PARAMETER PR=101
C
      INTEGER PA,PO
      INTEGER PR
      INTEGER CR,LP,PQ
      INTEGER ZERO
      INTEGER AXIS,BLNK,SMBL
      REAL MIN,MAX
C
      DIMENSION A(PA),O(PA,PO)
      DIMENSION SMBL(9)
      DIMENSION LINE(101)
C
      COMMON/A10/CR,LP,PQ
C
      DATA IMN,IMX/1,101/
      DATA ZERO/0/
      DATA AXIS/"+"/
      DATA BLNK/" "/
      DATA SMBL/"1","2","3","4","5","6","7","8","9"/
      DATA NS,PR/9,101/
C
11  FORMAT("0",10X,8X,1PG10.3,90X,1PG10.3,2X)
12  FORMAT("+",10X,10Y,.01A1)
13  FORMAT("+",15,5X,1PG9.2,1X)
14  FORMAT(" ")
C
      SLOPE=0 $IF(MIN.NE.MAX) SLOPE=(IMX-IMN)/(MAX-MIN)
C
      WRITE(LP,11) MIN,MAX
C
      DO 101 IP=1,PR,1
      LINE(IP)=AXIS
101  CONTINUE
      WRITE(LP,14)
      WRITE(LP,12) (LINE(IP),IP=1,PR,1)
C
      DO 111 IA=1,NA,1
      WRITE(LP,13) IA,A(IA)
      DO 112 IO=1,NO,1
      DO 113 IP=1,PR,1
      LINE(IP)=BLNK
113  CONTINUE
      IZ=SLOPE*(ZERO-MIN)+1 $LINE(IZ)=AXIS
      IS=SLOPE*(O(IA,IO)-MIN)+1 $LINE(IS)=SMBL(IO)
      WRITE(LP,12) (LINE(IP),IP=1,PR,1)
112  CONTINUE
      WRITE(LP,14)
111  CONTINUE
C

```

```

      RETURN
      END
C*****
      SUBROUTINE GRF
      +(A,O,PA,PO,NA,NC,MIN,MAX,TT,TX,TV,TH,PH,TS,PN,P,MP,MN,NM,MA,MO,MG)
C
C      CALCOMP
C
C      PARAMETER PB=512
C      PARAMETER PD=102
C      PARAMETER PS=5
C
      INTEGER PA,PO
      INTEGER PH,PN
      INTEGER CR,LP,PQ
      INTEGER PD,PO,PS
      INTEGER TT,TX,TV,TH
      INTEGER TS,ST
      INTEGER BFR
      INTEGER PP
      INTEGER ON,OO
      INTEGER PH,PV
      INTEGER SPC,SPS
      INTEGER SS,CS,SC
      INTEGER BOI,EOI,EOF
      REAL MIN,MAX
      REAL MOD
      LOGICAL LIN,LOG
C
      DIMENSION A(PA),O(PA,PO)
      DIMENSION TT(PM),TX(PM),TV(PM),TH(PM)
      DIMENSION TS(PN)
      DIMENSION BFR(512)
      DIMENSION DATA(102)
      DIMENSION SPS(5)
      DIMENSION XM(5),YM(5)
C
      COMMON/A10/CR,LP,PQ
C
      DATA PB/512/
      DATA PD/102/
      DATA PS/5/
      DATA XO,YO/+1.0,+3.5/
      DATA XV,YV/+0.0,+0.0/
      DATA XH,YH/+0.0,+3.0/
      DATA X4/+1.00,+1.00,+1.00,+1.00,+1.00/
      DATA YM/-0.50,-0.75,-1.00,-1.25,-1.50/
      DATA XT,YT/+1.00,-2.00/
      DATA XP,YP/+1.00,-2.25/
      DATA XN,YN/+2.00,-2.25/
      DATA XC,YC/+10.0,+0.0/
      DATA XI,YI/+0.0,+0.0/
      DATA XF,YF/+0.0,+0.0/
      DATA PP/00/
      DATA ON,OO/-3,+3/
      DATA PL/12.0/
      DATA HL,VL/+6.0,+6.0/
      DATA HA,VA/+0.0,+90.0/
      DATA HS,AS/+0.105,+0.0/
      DATA HF,AF/+0.105,+0.0/
      DATA HN,AN/+0.105,+0.0/
      DATA HP,AP/+0.105,+0.0/
      DATA OI/+3.0/
      DATA LW/10/

```

```

DATA LC/80/
DATA NT/8/
DATA SS/+1/
DATA CS,SC/-1,+10/
DATA SD/+1.0/
DATA ND/+2/
DATA NS/+5/
DATA SPS/0,1,2,3,4/
DATA BOI,EOI,EOF/1,-999,+999/

C
IF(MIN.EQ.MAX) RETURN

C
LINE=.FALSE. $IF((MA.EQ.0).AND.(MO.EQ.0)) LINE=.TRUE.
LOG=.FALSE. $IF((MA.NE.0).OR.(MO.NE.0)) LOG=.TRUE.

C
IF((MA.NE.0).AND.(MO.EQ.0)) LT=-1
IF((MA.NE.0).AND.(MO.NE.0)) LT= 0
IF((MA.EQ.0).AND.(MO.NE.0)) LT=+1

C
MB=MG $IF(MG.EQ.EOI) MB=BOI
IF(MB.EQ.BOI) CALL PLOTS(BFR,PB,PQ,PP)
IF(MB.EQ.BOI) CALL PLOT(XO,YO,ON)

C
CALL PLOTMX(PL)
IF(MA.EQ.0) CALL SCALE(A,HL,NA,SS)
IF(MA.NE.0) CALL LGSCAL(A,HL,NA,SS)
NC=NM*LW $IF(NC.GT.LC) NC=LC
PH=-NC $PV=+NC
AMN=ABS(MIN) $AMX=ABS(MAX) $MOD=AMAX1(AMN,AMX)
VTM=-MOD $HTM=A(NA*SS+1)
VDY=+MOD/OI $HDX=A(NA*SS+2)
IF(MO.EQ.0) CALL AXIS(XV,YV,TV,PV,VL,VA,VTM,VDY)
IF(MO.NE.0) CALL LGAXIS(XV,YV,TV,PV,VL,VA,VTM,VDY)
IF(MA.EQ.0) CALL AXIS(XH,YH,TH,PH,HL,HA,HTM,HDX)
IF(MA.NE.0) CALL LGAXIS(XH,YH,TH,PH,HL,HA,HTM,HDX)
CALL PLOT(XI,YI,OO)
DO 101 IO=1,NO,1
DO 102 IA=1,NA,1
DATA(IA)=O(IA,IO)
102 CONTINUE
DATA(NA+1)=VTM
DATA(NA+2)=VDY
SPC=SPS(IO)
IF(LINE) CALL LINE(A,DATA,NA,SS,NS,SPC)
IF(LOG) CALL LGLINE(A,DATA,NA,SS,NS,SPC,LT)
XS=XM(IO) $XD=XS+SD
YS=YM(IO) $YD=YS
ST=TS(IO)
NK=NM*LW $IF(NK.GT.LC) NK=LC
CALL SYMBOL(XS,YS,HS,SPC,AS,CS)
CALL SYMBOL(XD,YD,HS,ST,AS,SC)
101 CONTINUE
CALL SYMBOL(XT,YT,HT,TT,AF,NK)
IF(MP.NE.0) CALL NUMBER(XN,YN,HN,P,AN,ND)
IF(MP.NE.0) CALL SYMBOL(XP,YP,HP,TP,AP,NK)
CALL PLOT(XC,YC,ON)

C
ME=MG $IF(MG.EQ.EOI) ME=EOF
IF(ME.EQ.EOF) CALL PLOT(XF,YF,EOF)

C
RETURN
END
C***** A-39 *****
SUBROUTINE TPGX

```

```

      +(A,O,PA,PO,TT,IX,TV,TH,PM,DATA,PD,XI,XF,NX,MX,IXI,IXF,NXI)
C
C      TABLE/PLOT/GRAPH DATA
C
C      PARAMETER PN=5
C
      INTEGER PO
      INTEGER PM,PN
      INTEGER PA,PO
      INTEGER CR,LP,PQ
      INTEGER TT,IX,TV,TH
      INTEGER TS
      REAL    MIN,MAX
      COMPLEX DATA
C
      DIMENSION DATA(PD)
      DIMENSION A(PA),O(PA,PO)
      DIMENSION TT(PM),TX(PM),TV(PM),TH(PM)
      DIMENSION TS(5)
C
      COMMON/A10/CR,LP,PQ
      COMMON/B12/ML,RL,SL
      COMMON/B13/MIN,MAX,MD
      COMMON/B20/IN,NM
C
      1 FORMAT(8A10)
      2 FORMAT("0",8A10)
      3 FORMAT(10I10)
      4 FORMAT(2G10.0,I10)
      11 FORMAT("0","      INDEX"
      +      "      ARGUMENT"
      +      "      REAL"
      +      "      IMAGINARY"
      +      "      MAGNITUDE"
      +      "      ANGLE (RADIANS)"
      +      "      ANGLE (DEGREES)"/)
      12 FORMAT("0","1 -> REAL      "/"
      +      " "2 -> IMAGINARY    "/"
      +      " "3 -> MAGNITUDE     "/"
      +      " "4 -> ANGLE (RADIANS)"/
      +      " "5 -> ANGLE (DEGREES)"/)
C
      DATA PN/5/
      DATA TS
      +/"REAL      ","IMAGINARY ","MAGNITUDE ","ANGLE      ","ANGLE      "/"
      DATA P,MQ/0.0,0/
C
      READ(CR, 1) (TT(IT),IT=1,PM,1)
      READ(CR, 1) (TX(IT),IT=1,PM,1)
      READ(CR, 1) (TV(IT),IT=1,PM,1)
      READ(CR, 1) (TH(IT),IT=1,PM,1)
      READ(CR, 3)  MA,MO,MT,MP,MG
      READ(CR, 4)  MIN,MAX,MD
C
      IF((MT.EQ.0).AND.(MP.EQ.0).AND.(MG.EQ.0)) RETURN
      IF((MT.NE.0).OR.(MP.NE.0)) WRITE(LP, 2) (TT(IT),IT=1,PM,1)
C
      CALL AOZ(A,O,PA,PO,DATA,PD,ND,XI,XF,NX,MX,IXI,IXF,NXI)
C
      NA=ND $NO=5
      IF(MQ.NE.0) CALL SDB(O,PA,PO,NA,NC,ML,RL,SL)
      NA=ND $NO=5
      IF(MT.NE.0) WRITE(LP,11)
      IF(MT.NE.0) CALL TBL(A,O,PA,PO,NA,NO)

```

```

      NA=ND $NO=3
      CALL MNMX(3,PA,PO,NA,NO,XMN,XX,MIN,MAX,MO)
      NA=ND $NO=3
      IF(MP.NE.0) WRITE(LP,12)
      IF(MP.NE.0) CALL FLT(A,O,PA,PO,NA,NO,XMN,XX)
      NA=ND $NO=3
      IF(MG.NE.0) CALL GRF
      +(A,O,PA,PO,NA,NO,XMN,XX,TT,XX,TV,TH,PM,TS,PN,P,MQ,MN,NM,MA,MO,MG)
C
      RETURN
      END
C*****
      SUBROUTINE TPGK
      +(A,O,PA,PO,TT,XX,TV,TH,PM,DATA,PX,PY,
      + XI,XX,NX,MX,IXI,IXF,NXI,YI,YF,NY,MY,IYI,IYF,NYI)
C
C      GRAPH (TABLE/PLOT) (FREQUENCY DOMAIN)
C
      INTEGER PX,PY
      INTEGER PA,PO
      INTEGER PM
      INTEGER TT,XX,TV,TH
      COMPLEX DATA
C
      DIMENSION A(PA),O(PA,PO)
      DIMENSION TT(PM),XX(PM),TV(PM),TH(PM)
      DIMENSION DATA(PX,PY)
C
      DATA NQ/2/
C
      DO 100 IP=1,NQ,1
      IF(IP.EQ.1) CALL VPXY
      +(VI,VF,NV,MV,IYI,IVF,NVI,PI,PF,NP,MP,IPI,IPF,NPI,
      + XI,XX,NX,MX,IXI,IXF,NXI,YI,YF,NY,MY,IYI,IYF,NYI)
      IF(IP.EQ.2) CALL VPXY
      +(VI,VF,NV,MV,IYI,IVF,NVI,PI,PF,NP,MP,IPI,IPF,NPI,
      + YI,YF,NY,MY,IYI,IYF,NYI,XI,XX,NX,MX,IXI,IXF,NXI)
      CALL TPGC
      +(A,O,PA,PO,TT,XX,TV,TH,PM,DATA,PX,PY,IP,
      + VI,VF,NV,MV,IYI,IVF,NVI,PI,PF,NP,MP,IPI,IPF,NPI)
100 CONTINUE
C
      RETURN
      END
C*****
      SUBROUTINE TPGC
      +(A,O,PA,PO,TT,XX,TV,TH,PM,DATA,PX,PY,MZ,
      + XI,XX,NX,MX,IXI,IXF,NXI,YI,YF,NY,MY,IYI,IVF,NYI)
C
C      TABLE/PLOT/GRAPH DATA VS X OR Y
C
C      PARAMETER PN=5
C
      INTEGER PX,PY
      INTEGER PM
      INTEGER PA,PO
      INTEGER CH,LP,PD
      INTEGER TT,XX,TV,TH
      INTEGER TS
      INTEGER BOI,COI,EOI,FCF
      REAL MIN,MAX
      COMPLEX DATA
C
      DIMENSION DATA(PX,PY)

```

```

DIMENSION A(PA),O(PA,PO)
DIMENSION TT(PM),TX(PM),TV(PM),TH(PM)
DIMENSION TS(5)

C
COMMON/A10/CR,LP,FQ
COMMON/B12/ML,RL,SL
COMMON/B13/MIN,MAX,MD
COMMON/B20/MN,NM

C
DATA PN/5/
DATA TS
+/"REAL      ","IMAGINARY ","MAGNITUDE ","ANGLE      ","ANGLE      "/"
DATA M2/1/
DATA BOI,COI,EOI,EOF/1,2,-999,+999/

C
1 FORMAT(8A10)
2 FORMAT("0",3A10)
3 FORMAT(10I10)
4 FORMAT(2G10.0,I10)
11 FORMAT("0","      INDEX"
+      ,","      ARGUMENT"
+      ,","      REAL"
+      ,","      IMAGINARY"
+      ,","      MAGNITUDE"
+      ,","      ANGLE (RADIANS)"
+      ,","      ANGLE (DEGREES)"/")
12 FORMAT("0","1 -> REAL      "/"
+      " ", "2 -> IMAGINARY      "/"
+      " ", "3 -> MAGNITUDE      "/"
+      " ", "4 -> ANGLE (RADIANS)"/"
+      " ", "5 -> ANGLE (DEGREES)"/")
13 FORMAT("0","PARAMETER :",1PG20,10)

C
READ(CR, 1) (TT(IT),IT=1,PM,1)
READ(CR, 1) (TX(IT),IT=1,PM,1)
READ(CR, 1) (TV(IT),IT=1,PM,1)
READ(CR, 1) (TH(IT),IT=1,PM,1)
READ(CR, 3)  MA,MO,MT,MP,MG
READ(CR, 4)  MIN,MAX,MD

C
IF((MT.EQ.0).AND.(MP.EQ.0).AND.(MG.EQ.0)) RETURN
IF((MT.NE.0).OR.(MP.NE.0)) WRITE(LP, 2) (TT(IT),IT=1,PM,1)

C
DO 100 IP=IYI,IYF,NYI
MC=COI
IF((MG.EQ.BOI).AND.(IP.EQ.IYI)) MC=BOI
IF((MG.EQ.EOF).AND.(IP.EQ.IYF)) MC=EOF
IF((MG.EQ.EOI).AND.(IP.EQ.IYI)) MC=BOI
IF((MG.EQ.EOI).AND.(IP.EQ.IYF)) MC=EOF
IF((MG.EQ.EOI).AND.(IYI.EQ.IYF)) MC=EOI
CALL AOXY
+(A,O,PA,PO,DATA,PX,PY,NZ,P,IP,MZ,
+ XI,XF,NX,MX,IXI,IXF,NXI,YI,YF,NY,MY,IYI,IYF,NYI)
NA=NZ $NO=5
IF(MO.NE.0) CALL SDB(O,PA,PO,NA,NC,ML,RL,SL)
IF((MT.NE.0).OR.(MP.NE.0)) WRITE(LP,13) P
NA=NZ $NO=5
IF(MT.NE.0) WRITE(LP,11)
IF(MT.NE.0) CALL TBL(A,O,PA,PO,NA,NO)
NA=NZ $NO=3
CALL MNMX(O,PA,PO,NA,NO,XMN,XXM,MIN,MAX,MD)
NA=NZ $NO=3
IF(MP.NE.0) WRITE(LP,12)
IF(MP.NE.0) CALL PLT(A,O,PA,PO,NA,NO,XMN,XXM)

```

AD-A041 319

ROME AIR DEVELOPMENT CENTER GRIFFISS AFB N Y  
FAA LIGHTNING PROTECTION STUDY: LIGHTNING INDUCED TRANSIENTS ON--ETC(U)  
MAY 77 J D NORDGARD, C CHEN

F/G 9/1

DOT-FA72WAI-353

UNCLASSIFIED

FAA/RD-77-83

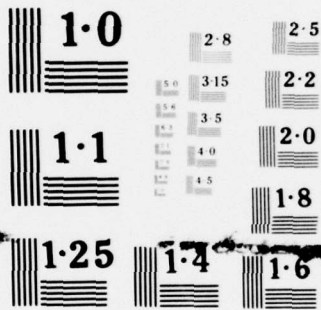
NL

2 OF 2  
ADA  
041319



END

DATE  
FILMED  
7-77



NATIONAL BUREAU OF STANDARDS  
MICROCOPY RESOLUTION TEST CHART

```

      NA=NZ $NO=3
      IF(MG.NE.0) CALL GRF
      +(A,O,PA,PO,NA,NC,XMN,XX,TT,TX,TV,TH,PM,TS,PN,P,MQ,MN,NM,MA,MO,MC)
100  CONTINUE
C
      RETURN
      END
C*****
      SUBROUTINE AOZ(A,O,PA,PO,DATA,PD,ND,XI,XF,NX,MX,IXI,IXF,NXI)
C
C      AESICCA/ORDINATE
C
      INTEGER PD
      INTEGER PA,PO
      INTEGER CR,LP,PQ
      COMPLEX DATA
C
      DIMENSION DATA(PD)
      DIMENSION A(PA),O(PA,PO)
C
      COMMON/A10/CR,LP,PQ
C
10  FORMAT("0","ERROR (AOZ) : PARAMETER LIMITS NOT SATISFIED"/)
C
      IF((NX.GT.PD).OR.(NX.GT.PA)) WRITE(LP,10)
C
      ID=IXF-IXI
      ND=ID/NXI+1
C
      IA=1
      DO 100 IX=IXI,IXF,NXI
      A(IA)=AF(XI,XF,NX,IX,MX)
      O(IA,1)= REAL(DATA(IX))
      O(IA,2)=AIMAG(DATA(IX))
      O(IA,3)= CABS(DATA(IX))
      AR=CA'NG(DATA(IX)) $AO=RODF(AR)
      O(IA,4)=AR
      O(IA,5)=AO
      IA=IA+1
100  CONTINUE
C
      RETURN
      END
C*****
      SUBROUTINE AOXY
      +(A,O,PA,PO,DATA,PX,PY,NZ,P,IP,YZ,
      + VI,VF,NV,MV,IVI,IVF,NVI,PI,PF,NP,MP,IP1,IFF,NP1)
C
C      AESICCA/ORDINATE
C
      INTEGER PX,PY
      INTEGER PA,PO
      INTEGER CR,LP,PQ
      COMPLEX DATA
      LOGICAL LVX,LPY
      LOGICAL LVY,LPX
      LOGICAL LVA
C
      DIMENSION DATA(PX,PY)
      DIMENSION A(PA),O(PA,PO)
C
      COMMON/A10/CR,LP,PQ
C
10  FORMAT("0","ERROR (TPGO) : PARAMETER LIMITS NOT SATISFIED"/)

```

```

C
LVX=(MZ.EQ.1).AND.(NV.GT.PX)
LPY=(MZ.EQ.1).AND.(NP.GT.PY)
LVY=(MZ.EQ.2).AND.(NP.GT.PX)
LPX=(MZ.EQ.2).AND.(NV.GT.PY)
LVA=(NV.GT.PA).OR.(NP.GT.PA)

C
IF(LVX.OR.LPY) WRITE(LP,10)
IF(LVY.OR.LPX) WRITE(LP,10)
IF(LVA) WRITE(LP,10)

C
ID=IVF-IVI
NZ=ID/NVI+1

C
P=AF(PI,PF,NP,IP,MP)

C
IA=1
DO 100 IV=IVI,IVF,NVI
A(IA)=AF(VI,VF,NV,IV,MV)
CALL XYVP(IX,IY,IV,IP,MZ)
O(IA,1)=REAL(DATA(IX,IY))
O(IA,2)=AIMAG(DATA(IX,IY))
O(IA,3)=CABS(DATA(IX,IY))
AR=CANG(DATA(IX,IY)) $AD=ROF(AR)
O(IA,4)=AR
O(IA,5)=AD
IA=IA+1
100 CONTINUE

C
RETURN
END

C*****
SUBROUTINE VPXY
+(VI,VF,NV,MV,IVI,IVF,NVI,PI,PF,NP,MP,IP1,IPF,NPI,
+ AI,AF,NA,MA,IAI,IAF,NAI,OI,OF,NO,MO,IOI,IOF,NOI)

C
C VARIABLE/PARAMETER
C
VI=AI $IVI=1 $PI=OI $IPI=IOI
VF=AF $IVF=NA $PF=OF $IPF=IOF
NV=NA $NVI=1 $NP=NO $NPI=NOI
MV=MA $MP=MO

C
RETURN
END

C*****
SUBROUTINE XYVP(IX,IY,IV,IP,MP)

C
C INDEX
C
IF(MP.EQ.1) IX=IV $IF(MP.EQ.2) IX=IP
IF(MP.EQ.1) IY=IP $IF(MP.EQ.2) IY=IV

C
RETURN
END

C*****
SUBROUTINE TLU(TX,TY,PT,NT,ND,X,Y,YD,MD,MX)

C
C TABLE LOOK UP (INTERPOLATION/EXTRAPOLATION/DIFFERENTIATION)
C
C PARAMETER DP=10
C
C INTEGER DP
C INTEGER PT

```

```

      INTEGER CR,LP,PQ
      COMPLEX TY
      COMPLEX QKMJ
      COMPLEX Q,QD
      COMPLEX Y,YD
C
      DIMENSION TX(PT),TY(PT)
      DIMENSION P(10),Q(10),QD(10)
C
      COMMON/A10/CR,LP,PQ
C
      DATA DP/10/
C
10  FORMAT("U","ERROR (TLU) : PARAMETER LIMITS NOT SATISFIED"/)
C
      IF(NT.GT.PT) WRITE(LP,10)
      IF(ND.GT.NT) WRITE(LP,10)
      IF(ND.GT.10) WRITE(LP,10)
      IF(ND.LE. 1) WRITE(LP,10)
C
      YD=0
      NTL=1
      NTU=NT
      NDH=ND/2
100  CONTINUE
      NTD=NTU-NTL $IF(NTD.LE.1) GO TO 110
      NTA=(NTL+NTU)/2
      IF(MX.LT.0) ASSIGN 101 TO IX
      IF(MX.GT.0) ASSIGN 102 TO IX
      GO TO IX,(101,102)
101  CONTINUE
      IF(X-TX(NTA)) 113,111,112
102  CONTINUE
      IF(X-TX(NTA)) 112,111,113
113  CONTINUE
      NTL=NTA $GO TO 100
112  CONTINUE
      NTU=NTA $GO TO 100
111  CONTINUE
      Y=TY(NTA) $IF(MD.EQ.0) RETURN
      NTL=NTA
110  CONTINUE
      NDL=NTL-NDH+1 $IF(NDU.GT.NT) NDL=NT-ND+1
      NDU=NTL+NDH+1 $IF(NDL.LT. 1) NDL=1
      DO 131 J=1,ND $K=NDL+J-1
      P (J)=TX(K)
      Q (J)=TY(K)
      QD(J)=0
131  CONTINUE
      I=ND-1
      DO 141 J=1,I $L=J+1
      DO 141 K=L,ND
      PKMJ=P(K)-P(J)
      QKMJ=Q(K)-Q(J)
      XMPJ=X -P(J)
      XMPK=X -P(K)
      Q (K)=(XMPJ*Q(K)-XMPK*Q(J))/PKMJ
      QD(K)=(QKMJ+XMPJ*QD(K)-XMPK*QD(J))/PKMJ
141  CONTINUE
      Y =Q (ND) $IF(X.EQ.TX(NTA)) Y=TY(NTA)
      YD=QD(ND)
C
      RETURN
      END

```

```

C*****
      COMPLEX FUNCTION IF(XI,XF,NX,MX,NP,FNCTN,X,Y,PW)
C
C      NEWTON/COTES PARABOLIC RULE
C
      INTEGER PW
      INTEGER CR,LP,PQ
      COMPLEX Y
      COMPLEX EVEN,ODD
      COMPLEX YI,YF
      COMPLEX YE,YO
      COMPLEX FNCTN
      COMPLEX I
      LOGICAL TEST
C
      DIMENSION X(PW),Y(PW)
C
      COMMON/A10/CR,LP,PQ
      COMMON/D21/XX,XXX
C
      10 FORMAT("0","ERROR (IF) : PARAMETER LIMITS NOT SATISFIED"//)
C
      CALL PARITY(NP,TEST)
C
      IF(NP.LT.5) WRITE(LP,10) $IF(TEST) WRITE(LP,10)
C
      I=0
      ML=-1
      NXM1=NX-1
      NPM1=NP-1
      NPM2=NP-2
      DO 100 IX=1,NXM1,1
      IM=IX
      IP=IX+1
      XM=AF(XI,XF,NX,IM,MX)
      XP=AF(XI,XF,NX,IP,MX)
      XO=(XP-XM)/NPM1
      YI=FNCTN(XX,XM,X,Y,PW)
      YF=FNCTN(XX,XP,X,Y,PW)
      EVEN=0
      ODD=0
      DO 101 IE=2,NPM1,2
      XE=AF(XM,XP,NP,IE,ML)
      YE=FNCTN(XX,XE,X,Y,PW)
      EVEN=EVEN+YE
101  CONTINUE
      DO 102 IO=3,NPM2,2
      XO=AF(XM,XP,NP,IO,ML)
      YO=FNCTN(XX,XO,X,Y,PW)
      ODD=ODD+YO
102  CONTINUE
      I=I+(XO/3)*(YI+4*EVEN+2*ODD+YF)
100  CONTINUE
      IF=I
C
      RETURN
      END
C*****
      SUBROUTINE PARITY(IT,TEST)
C
C      ODD/EVEN
C
C      LOGICAL TEST
C

```

```

      IC=(IT/2)*2
      IF(IC.EQ.IT) TEST=.TRUE.
      IF(IC.NE.IT) TEST=.FALSE.
C
      RETURN
      END
C*****
      COMPLEX FUNCTION QF(XI,XF,NX,MX,NQ,FNCTN,X,Y,PW)
C
C      GAUSS/LEGENDRE QUADRATURE
C
C      PARAMETER GL=5
C
      INTEGER GL
      INTEGER PW
      INTEGER CR,LP,PQ
      COMPLEX Y
      COMPLEX I
      COMPLEX TERM,SUM
      COMPLEX FNCTN
C
      DIMENSION X(PW),Y(PW)
      DIMENSION RL(5,5)
      DIMENSION WL(5,5)
C
      COMMON/410/CR,LP,PQ
      COMMON/021/XX,XXX
C
      DATA GL/5/
      DATA RL
      +/0.00000000,-0.57735027,-0.77459667,-0.86113631,-0.90617984,
      + 0.00000000,+0.57735027,+0.00000000,-0.33998104,-0.53846931,
      + 0.00000000,+0.00000000,+0.77459667,+0.33998104,+0.00000000,
      + 0.00000000,+0.00000000,+0.00000000,+0.86113631,+0.53846931,
      + 0.00000000,+0.00000000,+0.00000000,+0.00000000,+0.90617984/
      DATA WL
      +/2.00000000,+1.00000000,+0.55555556,+0.34785484,+0.23692689,
      + 0.00000000,+1.00000000,+0.88888889,+0.65214514,+0.47862869,
      + 0.00000000,+0.00000000,+0.55555556,+0.65214514,+0.56888889,
      + 0.00000000,+0.00000000,+0.00000000,+0.34785484,+0.47862869,
      + 0.00000000,+0.00000000,+0.00000000,+0.00000000,+0.23692689/
C
10  FORMAT("D","ERROR (QF) : PARAMETER LIMITS NOT SATISFIED"/)
C
      IF(NQ.GT.GL) WRITE(LP,10)
C
      I=0
      NXM1=NX-1
      DO 101 IX=1,NXM1,1
        IM=IX
        IP=IX+1
        XM=AF(XI,XF,NX,IM,MX)
        XP=AF(XI,XF,NX,IP,MX)
        XA=(XP-XM)/2
        XC=(XP+XM)/2
        SUM=0
        DO 102 IQ=1,NQ,1
          XS=XC+RL(NQ,IQ)*XA
          TERM=WL(NQ,IQ)*FNCTN(XX,XS,X,Y,PW)
          SUM=SUM+TERM
102  CONTINUE
          I=I+XA*SUM
101  CONTINUE
      QF=I

```

```

C      RETURN
C      END
C*****
C      REAL FUNCTION NRF(X,FF,FDF)
C
C      NEWTON/RAPHSON
C
C      INTEGER CR,LP,PQ
C
C      COMMON/410/CR,LP,PQ
C      COMMON/311/IMX,EMX
C
C      10 FORMAT("0","ERROR (NRF) : MAXIMUM ITERATION LIMIT EXCEEDED"/)
C
C      XM=X
C      DO 101 II=1,IMX,1
C      F=FF(XM)
C      FD=FDF(XM)
C      XP=XM/2 $IF(FD.NE.0) XP=XM-F/FD
C      DIF=ABS(XP-XM)
C      ERR=DIF $IF(XM.NE.0) ERR=DIF/XM
C      IF(ERR.LT.EMX) GO TO 100
C      XM=XP
C 101 CONTINUE
C      WRITE(LP,10)
C 100 CONTINUE
C      NRF=XP
C
C      RETURN
C      END
C*****
C      SUBROUTINE SRM(KN,N,RTS,MAXIT,EP1,EP2,FN,FNREAL)
C
C      MULLER METHOD
C      HENRICI,"ELEMENTS OF NUMERICAL ANALYSIS",PP. 198-2
C      MULLER,"A METHOD OF SOLVING ALGEBRAIC EQUATIONS"
C      MATH TABLES AND AIDS TO COMPUTATION 10 (1956), PP. 208-215
C
C      COMPLEX RTS(1)
C      LOGICAL FNREAL
C      COMPLEX X0,X1,X2,FX0,FX1,FX2,RT,FRT,FPRT,DEN
C      COMPLEX A2,B2,C2,D2,E2,F2,H2,Q2,Q21
C      EPS1=EP1
C      EPS2=EP2
C      IF(EPS1.LE.0.0) EPS1=1.0E-12
C      IF(EPS2.LE.0.0) EPS2=1.0E-20
C      KNL=KN
C      NL=KN+N
C 10 CONTINUE
C      IF(KNL.LT.NL) GO TO 20
C      RETURN
C 20 KNL=KNL+1
C      KOUNT=0
C 30 X0=RTS(KNL)-0.5
C      ASSIGN 40 TO NN
C      RT=X0
C      GO TO 40
C 40 FX0=FPRT
C      X1=RTS(KNL)+0.5
C      ASSIGN 50 TO NN
C      RT=X1
C      GO TO 40
C 50 FX1=FPRT

```

```

X2=RTS(KNL)
ASSIGN 60 TO NN
RT=X2
GO TO 30
60 FX2=FPRT
H2=X2-X1
Q2=(X2-X1)/(X1-X0)
65 CONTINUE
Q21=Q2+1.0
A2=Q2*(FX2-Q21*FX1+Q2*FX0)
B2=(Q21+Q2)*FX2-Q21*Q21*FX1+Q2*Q2*FX0
C2=Q21*FX2
D2=B2*B2-4.0*A2*C2
IF((FNREAL).AND.(REAL(D2).LT.0.0)) D2=0.0
D2=CSQRT(D2)
E2=B2+D2
F2=B2-D2
IF(CABS(F2).GT.CABS(E2)) E2=F2
IF(CABS(E2).GT.0.0) GO TO 70
E2=1.0
70 CONTINUE
Q2=-2.0*C2/E2
FX0=FX1
FX1=FX2
H2=H2*Q2
RT=RT+H2
ASSIGN 75 TO NN
GO TO 30
75 CONTINUE
FX2=FPRT
IF(CABS(H2).LT.EPS1*CABS(RT)) GO TO 100
IF((CABS(FRT).LT.EPS2).AND.(CABS(FPRT).LT.EPS2)) GO TO 100
IF(KOUNT.GT.MAXIT) GO TO 100
IF(CABS(FX2).LT.10.0*CABS(FX1)) GO TO 65
H2=0.5*H2
Q2=0.5*Q2
FR=FR-H2
80 CONTINUE
KOUNT=KOUNT+1
CALL FN(RT,FRT)
FPRT=FRT
IF(KNL.LT.2) GO TO 91
DO 90 I=2,KNL
DEN=RT-RTS(I-1)
IF(CABS(DEN).LT.EPS2) GO TO 92
90 FPRT=FPRT/DEN
91 CONTINUE
GO TO NN,(40,50,60,75)
100 CONTINUE
RTS(KNL)=RT
GO TO 10
42 RT=RT+0.001
GO TO 30
END
C*****
C      SUBROUTINE BNHF(ORD,AFG,BF,NE,HFM,HFP)
C
C      BESSEL  FUNCTIONS
C      NEUMANN FUNCTIONS
C      HANKEL  FUNCTIONS
C
C      INTEGER ORD,OA
C      REAL    MOD
C      COMPLEX CI

```

```

COMPLEX ARG
COMPLEX BF,NF,HFM,HFP
C
DATA POLE/1.0E+39/
DATA BPL/0.001/
DATA BPU/1E.0/
C
CI=CMPLX(0.0,1.0)
C
OA=IABS(ORD)
MOD=CABS(ARG)
C
BF=+0 $IF(OA.EQ.0) BF=+1
NF=-POLE
HFM=BF-CI*NF
HFP=BF+CI*NF
C
IF (MOD.LE.0) RETURN
IF (MOD.LT.BPL) CALL SRAS(OA,ARG,BF,NF,HFM,HFP)
IF ((MOD.GE.BPL).AND.(MOD.LE.BPU)) CALL SRRS(OA,ARG,BF,NF,HFM,HFP)
IF (MOD.GT.BPU) CALL SPAL(OA,ARG,BF,NF,HFM,HFP)
C
IF(ORD.GT.0) RETURN
C
BF=((-1)**OA)*BF
NF=((-1)**OA)*NF
HFM=BF-CI*NF
HFP=BF+CI*NF
C
RETURN
END
C*****
SUBROUTINE SRAS(ORD,ARG,BF,NF,HFM,HFP)
C
ASYMPTOTIC EXPANSION (SMALL ARGUMENTS)
C
INTEGER ORD
COMPLEX CI
COMPLEX ARG,AD2
COMPLEX BF,NF,HFM,HFP
C
DATA PI/3.14159263/
C
CI=CMPLX(0.0,1.0)
C
AD2=ARG/2
C
FP=FF(ORD)
FM=FF(ORD-1)
C
BF=+(AD2**(+ORD))/FP
NF=- (AD2**(-ORD))*(FM/PI) $IF(ORD.EQ.0) NF=(2/PI)*CLOG(ARG)
HFM=BF-CI*NF
HFP=BF+CI*NF
C
RETURN
END
C*****
SUBROUTINE SRRS(ORD,ARG,BF,NF,HFM,HFP)
C
ASCENDING SERIES
C
INTEGER ORD,OM1,OP1
COMPLEX CI

```

```

COMPLEX ARG,AD2
COMPLEX TM,TP
COMPLEX SB,SN,SY
COMPLEX BSF,NSF,YSF
COMPLEX YF,NFM,NFP
COMPLEX BF,NF,HFM,HFP

C
COMMON/B11/IMX,EMX

C
DATA PI/3.14159263/

C
CI=CMPLX(0.0,1.0)

C
OM1=ORD-1
OP1=ORD+1

C
AD2=ARG/2 $TM=AD2**(-ORD) $TP=AD2**(ORD)

C
FO =FF(ORD) $G1 =GF( 1)
FOM1=FF(OM1) $GOP1=GF(OP1)
GO=G1+GOP1

C
SB=BSF(ORD,ARG)
SN=NSF(ORD,ARG,GO)
SY=YSF(ORD,ARG)

C
BF=(TP/FO)*SB
YF=(2/PI)*CLOG(AD2)*BF
NFM=-(TM/PI)*FOM1*SY
NFP=-(TP/PI)*(1/FO)*SN
NF=YF+NFM+NFP
HFM=BF-CI*NF
HFP=BF+CI*NF

C
RETURN
END
C*****
SUBROUTINE SRAL(ORD,ARG,BF,NF,HFM,HFP)
C
C ASYMPTOTIC EXPANSION (LARGE ARGUMENTS)
C
C
INTEGER ORD
COMPLEX ARG
COMPLEX CI
COMPLEX CC
COMPLEX X,C
COMPLEX CX,SX
COMPLEX AMX,APX
COMPLEX EMX,EPX
COMPLEX P,Q
COMPLEX BF,NF,HFM,HFP
COMPLEX DCEXP

C
DATA PI/3.14159263/

C
CI=CMPLX(0.0,1.0)

C
X=ARG-(2*ORD+1)*(PI/4) $CX=CCOS(X) $SX=CSIN(X)
CC=2/(PI*ARG) $C=CSQRT(CC)

C
AMX=-CI*X $LMX=DCEXP(AMX)
APX=+CI*X $EPX=DCEXP(APX)
C
CALL SRFD(ORD,ARG,P,Q)

```

```

C      BF=C*(P*CX-1)*SX)
C      NF=C*(P*SX+Q*CX)
C      HFM=C*(P-CI*Q)*EMX
C      HFP=C*(P+CI*Q)*EPX
C
C      RETURN
C      END
C*****
C      SUBROUTINE SRPQ(ORD,ARG,P,Q)
C
C      HANKEL ASYMPTOTIC EXPANSIONS
C
C      INTEGER CR,LP,PQ
C      INTEGER ORD,OM
C      INTEGER SGN
C      COMPLEX ARG,AT8
C      COMPLEX T,S
C      COMPLEX P,Q
C
C      COMMON/A11/CR,LP,PQ
C      COMMON/B11/IMX,EMX
C
C      10 FORMAT("0","ERROR (SRPQ) : MAXIMUM ITERATION LIMIT EXCEEDED"/)
C
C      P=1 $EP=1
C      Q=0 $EQ=1
C      T=1
C
C      OM=4*ORD*ORD
C      AT8=3*ARG
C
C      DO 101 IX=1,IMX,2 $IXP1=IX+1
C      DO 102 IY=IX,IXP1,1
C      SGN=(-1)**IY
C      IO=2*IY-1 $IS=IO*IO
C      S=SGN*((OM-IS)/(IY*AT8))
C      T=T*S
C      IF(SGN.EQ.-1) P=P+T
C      IF(SGN.EQ.+1) Q=Q+T
C      102 CONTINUE
C      IF(CABS(P).NE.0) EP=CABS(T/P)
C      IF(CABS(Q).NE.0) EQ=CABS(T/Q)
C      ERR=AMAX1(EP,EQ) $IF(ERR.LT.EMX) GO TO 100
C      101 CONTINUE
C      WRITE(LP,10)
C      100 CONTINUE
C
C      RETURN
C      END
C*****
C      SUBROUTINE MBNHF(CRD,ARG,MBF,MNF,MHFM,MHFP)
C
C      MODIFIED BESSEL FUNCTIONS
C      MODIFIED NEUMANN FUNCTIONS
C      MODIFIED HANKEL FUNCTIONS
C
C      INTEGER ORD
C      COMPLEX CI
C      COMPLEX ARG,CA
C      COMPLEX BF,NF,HFM,HFP
C      COMPLEX MBF,MNF,MHFM,MHFP
C
C      DATA PI/3.14159263/

```

```

C      CI=CMPLX(0.0,1.0)
C      CA=CI*ARG $CALL BNHF(ORD,CA,BF,NF,HFM,HFP)
C      MBF=(CI**(-ORD))*BF
C      MNF=NF
C      MHFM=(PI/2)*(CI** (ORD+1))*HFM
C      MHFP=(PI/2)*(CI** (ORD+1))*HFP
C      RETURN
C      END
C*****
C      COMPLEX FUNCTION BSF(ORD,ARG)
C      BESSEL SERIES
C      INTEGER CR,LP,PQ
C      INTEGER ORD
C      COMPLEX ARG,AD2
C      COMPLEX TC,TR,TB,SB
C
C      COMMON/A10/CR,LP,PQ
C      COMMON/B11/IMX,EMX
C      10 FORMAT("D","ERROR (BSF) : MAXIMUM ITERATION LIMIT EXCEEDED"//)
C      AD2=ARG/2 $TC=AD2**2
C      TB=1 $SB=1 $EB=1
C
C      DO 101 IX=1,IMX,1
C      TR=-TC/(IX*(IX+ORD))
C      TB=TB*TR
C      SB=SB+TB
C      IF (CABS(SB).NE.0) EP=CABS(TB/SB) $IF(EB.LT.EMX) GO TO 100
101  CONTINUE
C      WRITE(LP,10)
100  CONTINUE
C      BSF=SB
C      RETURN
C      END
C*****
C      COMPLEX FUNCTION NSF(ORD,ARG,GO)
C      NEUMAN SERIES
C      INTEGER CR,LP,PQ
C      INTEGER ORD
C      COMPLEX ARG,AD2
C      COMPLEX TS,TP
C      COMPLEX TC,TR,TX,TN,SN
C
C      COMMON/A10/CR,LP,PQ
C      COMMON/B11/IMX,EMX
C      10 FORMAT("D","ERROR (NSF) : MAXIMUM ITERATION LIMIT EXCEEDED"//)
C      AD2=ARG/2 $TC=AD2**2
C
C      TS=GO $SN=TS $EN=1
C      TP=1
C      TN=TP

```

```

C
DO 101 IX=1,IMX,1
TX=+FLOAT(1)/IX+FLOAT(1)/(IX+ORD)
TS=TS+TX
TR=-TC/(IX*(IX+ORD))
TP=TP*TR
TN=TS*TP
SN=SN+TN
IF(CABS(SN).NE.0) EN=CABS(TN/SN) $IF(EN.LT.EMX) GO TO 100
101 CONTINUE
WRITE(LP,10)
100 CONTINUE
NSF=SN
C
RETURN
END
C*****
COMPLEX FUNCTION YSF(ORD,ARG)
C
C NEUMANN SERIES
C
C INTEGER ORD,OM1
C COMPLEX ARG,AD2
C COMPLEX TC,TR,TY,SY
C
YSF=0 $IF(ORD.EQ.0) RETURN
YSF=1 $IF(ORD.EQ.1) RETURN
C
OM1=ORD-1
C
AD2=ARG/2 $TC=AD2**2
C
TY=1 $SY=1
C
DO 100 IX=1,OM1,1
TR=+TC/(IX*(ORD-IX))
TY=TY*TR
SY=SY+TY
100 CONTINUE
YSF=SY
C
RETURN
END
C*****
REAL FUNCTION FF(IR)
C
C FACTORIAL
C
FF=1 $IF(IR.LT.1) RETURN
C
F=1
DO 111 IX=1,IP,1
F=F*IX
111 CONTINUE
FF=F
C
RETURN
END
C*****
REAL FUNCTION GF(IR)
C
C GAMMA
C
DATA GAMMA/0.577215664901533/

```

```

C
GF=-GAMMA $IF(IR.EQ.1) RETURN
C
SUM=0
IRM1=IR-1
DO 100 IX=1,IRM1,1
TERM=FLOAT(1)/IX
SUM=SUM+TERM
100 CONTINUE
GF=-GAMMA+SUM
C
RETURN
END
C*****
REAL FUNCTION CANG(C)
C
C COMPLEX ANGLE
C
REAL R,I
COMPLEX C
C
CANG=0
R=REAL(C) $IF(R.EQ.0) RETURN
I=AIMAG(C)
CANG=ATAN2(I,R)
C
RETURN
END
C*****
COMPLEX FUNCTION DCEXP(C)
C
C DOUBLE PRECISION COMPLEX EXPONENTIAL
C
DOUBLE PRECISION DR,DI
DOUBLE PRECISION DE,DC,DS
C
COMPLEX CC
COMPLEX C
COMPLEX EXP
C
CC=CMPLX(1.0,1.0)
C
CR=REAL(C) $DR=DBLE(CR)
CI=AIMAG(C) $DI=DBLE(CI)
C
DE=DEXP(DR) $SE=SNGL(DE)
DC=DCOS(DI) $SC=SNGL(DC)
DS=DSIN(DI) $SS=SNGL(DS)
C
EXP=SE*(SC+DC*SS) $DCEXP=EXP
C
RETURN
END
C*****
SUBROUTINE FTDR(DU,DATA,PX,PY)
C
C FASTRAN DRUM READ
C
C
INTEGER PX,PY
INTEGER CY,LP,PQ
INTEGER DU
INTEGER DMN
INTEGER RD
INTEGER END,EUF

```

```

      INTEGER RWND, WAIT
      COMPLEX DATA
C
      DIMENSION DATA(PX, PY)
C
      COMMON/A10/CR, LP, PQ
C
      DATA RWND/10/
      DATA RD / 2/
      DATA WAIT/22/
C
10  FORMAT("U", "READ  TRANSFER COMPLETE / DRUM :", I10)
11  FORMAT("U", "READ  TRANSFER ERROR   / UNIT  :", I10/
      +      " ", " ", " " / END   :", I10/
      +      " ", " ", " " / EOF   :", I10)
C
      DMN=PX*PY $END=DMN $EOF=DMN
C
      CALL NTRAN(DU, RWND)
      CALL NTRAN(DU, RD ,END, DATA, EOF)
      CALL NTRAN(DU, WAIT)
C
      READ(DU) DATA
C
      IF(EOF.EQ.END) WRITE(LP, 10) DU
      IF(EOF.NE.END) WRITE(LP, 11) DU, END, EOF
C
      RETURN
      END
C*****
      SUBROUTINE FTDW(DU, DATA, PX, PY)
C
      FASTRAN DRUM WRITE
C
      INTEGER PX, PY
      INTEGER CR, LP, PQ
      INTEGER DU
      INTEGER DMN
      INTEGER WRT
      INTEGER END, EOF
      INTEGER RWND, WAIT
      COMPLEX DATA
C
      DIMENSION DATA(PX, PY)
C
      COMMON/A10/CR, LP, PQ
C
      DATA RWND/10/
      DATA WRT / 1/
      DATA WAIT/22/
C
10  FORMAT("U", "WRITE TRANSFER COMPLETE / DRUM :", I10)
11  FORMAT("U", "WRITE TRANSFER ERROR   / UNIT  :", I10/
      +      " ", " ", " " / END   :", I10/
      +      " ", " ", " " / EOF   :", I10)
C
      DMN=PX*PY $END=DMN $EOF=DMN
C
      CALL NTRAN(DU, RWND)
      CALL NTRAN(DU, WRT, END, DATA, EOF)
      CALL NTRAN(DU, WAIT)
C
      WRITE(DU) DATA
C

```

```

      IF(EOF.EQ.END) WRITE(LP,10) DU
      IF(EOF.NE.END) WRITE(LP,11) DU,END,EOF
C
      RETURN
      END
C*****
      REAL FUNCTION RDF(AR)
C
C      RADIANT/DEGREE
C
      DATA PI/3.14159265/
C
      AR=AR*(180/PI) $RDF=AR
C
      RETURN
      END
C*****
      REAL FUNCTION DRF(AD)
C
C      DEGREE/RADIANT
C
      DATA PI/3.14159265/
C
      AR=AD*(PI/180) $DRF=AR
C
      RETURN
      END

```

	100	1.0E-10	
	11.0	20.0	
0.0	1.0		-1
	+3	0	+1
	4	5	
	6	3	
	1	1	1
	0	0	0
	0	1	1
	0		
	1		
	100.0E-06	1.0E+06	
	51	51	
1000.0	1000.0	1000.0	
	1.0E+030.5	0.5	0
	20.0E+030.5	0.5	0
	1.0E+030.5	0.5	0
0.0	1.0E-06	100.0E-06	1.0E-06
0.0	1.0E-06	100.0E-06	1.0E-06
0.0	1.0E-06	100.0E-06	40.0E-03
	-2.000E+03+2.000E+03	41	
	-0.000E+03+1.000E+03	41	
	-0.000E+03+1.000E+03	3	
	1.0E+00	+1.0E+06	61
	0.0E-03	1.0E-03	61
	-1	-1	-1
	1	41	10
	1	41	10
	1	3	1
	1	61	20
	1	61	10
	0.913E-03		
	2.953E-03	3.953E-03	
	4.953E-03	5.953E-03	
	6.953E-03		
1.0			
0.0			
1.0	5.0	1.5E-03	
1.0	2.26	1.0E-03	
1.0	1.0	5.7E+07	
1.0	2.26	1.0E-03	
1.0	1.0	5.7E+07	
1.0	2.26	1.0E-03	
1.0	1.0	5.7E+07	
50.0	0.0		
50.0	0.0		
	1		
	1		

# APPENDIX B Sample Input Data Listing

IMX,EMX  
ML,RL,SL  
MIN,MAX  
NO,MO,MX  
NQ,NI  
MN,NM  
MPS  
RPS  
WPS  
MR  
MP  
TM,WM  
NK,NC  
CG  
PS  
RX  
RS  
PS  
RX  
RS  
EI,EF,NE  
XI,XF,NX  
YI,YF,NY  
WI,WF,NW  
TI,TF,NT  
MEXYWT  
IEIFN  
IXIFN  
IYIFN  
IWIFN  
ITIFN  
RW  
RSM,RSP  
RAM,RAP  
RC  
DO  
SO  
RUEOE  
RUEOI  
RUEOA  
RUEO2  
RUEOS  
RUEO1  
RUEOW  
ZL1  
ZL2  
MIO  
MLC

FIGURE 1 : LIGHTNING PULSE (TIME DOMAIN) VS TIME

NA  
PULSE (A)  
TIME (S)

	0	0	1	1	1
0.0		20.0E+03	-1		

FIGURE 2 : LIGHTNING PULSE (FREQUENCY DOMAIN) VS FREQUENCY

NA  
PULSE (A)  
FREQUENCY (RAD)

	1	0	1	1	2
-1.42	+2.36	-1			

FIGURE 3A : OUTER EXCITATION (FREQUENCY DOMAIN) VS POSITION

FREQUENCY  
EXCITATION (V/M)  
POSITION (M)

HTPGK

HTPGC

0	0	1	1	2	MTPGX E
-1.15E-06	+1.22E-03	-1			
FIGURE 3B : OUTER EXCITATION (FREQUENCY DOMAIN) VS FREQUENCY					
POSITION					
EXCITATION (V/M)					
FREQUENCY (RAD)					
1	0	0	0	0	MTPGW E
-1.15E-06	+1.22E-03	-1			
FIGURE 4A : OUTER CURRENT (FREQUENCY DOMAIN) VS POSITION					
FREQUENCY					
CURRENT (A)					
POSITION (M)					
0	0	1	1	2	MTPGX O
-0.0396	+0.146	-1			
FIGURE 4B : OUTER CURRENT (FREQUENCY DOMAIN) VS FREQUENCY					
POSITION					
CURRENT (A)					
FREQUENCY (RAD)					
1	0	0	0	0	MTPGW O
-0.0395	+0.146	-1			
FIGURE 5 : TRANSFER RATIO (FREQUENCY DOMAIN) VS FREQUENCY					
NA					
TRANSFER RATIO					
FREQUENCY (RAD)					
1	0	1	1	2	MTPGW T
-0.0290	+0.0358	-1			
FIGURE 6A : INNER EXCITATION (FREQUENCY DOMAIN) VS POSITION					
FREQUENCY					
EXCITATION (V/M)					
POSITION (M)					
0	0	1	1	2	MTPGX E
-1.15E-06	+1.22E-03	-1			
FIGURE 6B : INNER EXCITATION (FREQUENCY DOMAIN) VS FREQUENCY					
POSITION					
EXCITATION (V/M)					
FREQUENCY (RAD)					
1	0	0	0	0	MTPGW E
-1.15E-06	+1.22E-03	-1			
FIGURE 7A : INNER CURRENT (FREQUENCY DOMAIN) VS POSITION					
FREQUENCY					
CURRENT (A)					
POSITION (M)					
0	0	0	0	0	MTPGX I
-0.00189	+0.00522	-1			
FIGURE 7B : INNER CURRENT (FREQUENCY DOMAIN) VS FREQUENCY					
POSITION					
CURRENT (A)					
FREQUENCY (RAD)					
1	0	1	1	2	MTPGW I
-0.00189	+0.00522	-1			
FIGURE 7C : INNER VOLTAGE (FREQUENCY DOMAIN) VS POSITION					
FREQUENCY					
VOLTAGE (V)					
POSITION (M)					
0	0	0	0	0	MTPGX I
-0.00560	+0.0163	-1			
FIGURE 7D : INNER VOLTAGE (FREQUENCY DOMAIN) VS FREQUENCY					
POSITION					
VOLTAGE (V)					
FREQUENCY (RAD)					
1	0	1	1	2	MTPGW I
-0.00560	+0.0163	-1			
FIGURE 8A : INNER CURRENT (TIME DOMAIN) VS POSITION					
TIME					

CURRENT (A)

POSITION (M)

0 0 0 0 0  
-0.0 +12.3 -1

FIGURE 8B : INNER CURRENT (TIME DOMAIN) VS TIME

POSITION

CURRENT (A)

TIME (S)

0 0 1 1 2  
-0.0 +12.3 -1

FIGURE 8C : INNER VOLTAGE (TIME DOMAIN) VS POSITION

TIME

VOLTAGE (V)

POSITION (M)

0 0 0 0 0  
-0.333 +39.6 -1

FIGURE 8D : INNER VOLTAGE (TIME DOMAIN) VS TIME

POSITION

VOLTAGE (V)

TIME (S)

0 0 1 1 +999  
-0.333 +39.6 -1

MTPGX T

MTPGT T

MTPGX T

MTPGT T

APPENDIX C

FAA Lightning Study Participants

Florida Institute of Technology - Cable Testing

Dr. Andrew W. Revay, Jr.  
Richard M. Cosel

Lightning and Transient Research Institute - High Voltage Facility

James C. Stahmann

Purdue University - Protective Devices

Warren Peele (Project Leader)  
Dr. Chin-Lin Chen

Georgia Institute of Technology - Equipment Analysis

Keith Huddleston  
Dr. Ronal Larson  
Dr. John Nordgard

Air Force Institute of Technology - Reliability Aspects

Professor T. L. Regulinski

Augustana Research Institute

Dr. Jerry L. Hanson

Table 1. List of Input Cards and Formats

<u>Card</u>	<u>J</u>	<u>Label</u>	<u>Format</u>
1		IMX,JMX,EMX	2110,G10.0
2		ML,RL,SL	110,2G10.0
3		MIN,MAX,MP	2G10.0,110
4		ND,MD,MXX	3110
5		NQ,NI	2110
6		IFT,JFT	2110
7		NM,MN	2110
8		MLS,MCO,MTT,MCI,MTL,MFT	6110
9		RLS,RCO,RTT,RCI,RTL,RFT	6110
10		WLS,WCO,WTT,WCI,WTL,WFT	6110
11		MR,MG	2110
12		NR,IR	2110
13		TPK	G10.0
14		EPK	G10.0
15		MQ	110
16		TM,WM	2G10.0
17		NK,NC	2110
18		PGPS,PGPX,PGRS	3G10.0
19		PSPS,DCPS,DKPS,MPS,NPS	3G10.0,2110
20		PSRX,DCRX,DKRX,MRX,NRX	3G10.0,2110
21		PSRS,DCRS,DKRS,MRS,NRS	3G10.0,2110
22		TSPS,TRPS,TDPS,TPPS,TCPS	5G10.0
23		TSRX,TRRX,TDRX,TPRX,TCRX	5G10.0
24		TSRS,TRRS,TDRS,TPRS,TCRS	5G10.0
25		SI,SF,NS	2G10.0,110
26		XI,XF,NX	2G10.0,110
27		YI,YF,NY	2G10.0,110
28		WI,WF,NW	2G10.0,110
29		TI,TF,NT	2G10.0,110
30		MS,MX,MX,MW,MT	5110
31		IEI,IEF,NEI	3110
32		IXI,IXF,NXI	3110
33		IYI,IYF,NYI	3110
34		IWI,IWF,NWI	3110
35		ITI,ITF,NTI	3110
36		RW	G10.0
37		RSM,RSP	2G10.0
38		RAM,RAP	2G10.0
39		RS	G10.0
40		ZO	G10.0
41		YO	G10.0
42		URE,ERE,OE	3G10.0
43		URI,ERI,OI	3G10.0
44		URA,ERA,OA	3G10.0
45		UR2,ER2,O2	3G10.0
46		URS,ERS,OS	3G10.0
47		URI,ERI,OI	3G10.0
48		URW,ERW,OW	3G10.0
49		ZLM	2G10.0
50		ZLP	2G10.0

<u>Numonic</u>	<u>Type</u>	<u>Description</u>	<u>Numonic</u>	<u>Type</u>	<u>Description</u>
IMX	Integer	MAXIMUM ITERATION	MG	integer	MODE GUESS
JMX	integer	MAXIMUM DIGITS	NP	integer	ROOT NUMBER
EMX	real	MAXIMUM ERROR	IR	integer	ROOT INDEX
ML	integer	LOG MODE	TRK	real	ROOT GUESS
RL	real	LOG RATIO	EFL	real	ROOT ERROR
SL	real	LOG SCALE	TM	real	MULTIPLIER
MTN	real	PLOT MINIMUM	WM	real	MULTIPLIER
MAX	real	PLOT MAXIMUM	NK	integer	NUMBER OF PULSES
MP	integer	PLOT MODE	NC	integer	NUMBER OF PULSES
ND	integer	POLYNOMIAL DEGREE	PGPS	real	PULSE GUESS PRESTRIKE
MD	integer	CALCULATE DERIVATIVE	PGRX	real	PULSE GUESS RETURN
MXX	integer	TABLE ORDER	PGRS	real	PULSE GUESS RESTRIKE
NQ	integer	QUADRATURE GRID	PSFS	real	PULSE PEAK PRESTRIKE
NI	integer	INTEGRATION INDEX	DCPS	real	DECAY CONSTANT PRESTRIKE
IFT	integer	INDEX FOURIER TRANSFORM	DKPS	real	DECAY CONSTANT PRESTRIKE
JFT	integer	INDEX FOURIER TRANSFORM	MPS	integer	MODE DECAY PRESTRIKE
NM	integer	TITLE NUMBER	NPS	integer	MODE NUMBER PRESTRIKE
MN	integer	MODE NUMBER	PSRX	real	PULSE PEAK RETURN
MLS	integer	MODE LIGHTNING STROKE	DCRX	real	DECAY CONSTANT RETURN
MCO	integer	MODE COUPLE (OUTER)	DKRX	real	DECAY CONSTANT RETURN
MTT	integer	MODE TRANSFER	MRX	integer	MODE DECAY RETURN
MCI	integer	MODE COUPLE (INNER)	NRX	integer	MODE NUMBER RETURN
MTL	integer	MODE TRANSMISSION LINE	PSRS	real	PULSE PEAK RESTRIKE
TFT	integer	MODE INVERSE	DCRS	real	DECAY CONSTANT RESTRIKE
RLS	integer	READ STROKE	DKRS	real	DECAY CONSTANT RESTRIKE
RCO	integer	READ COUPLE (OUTER)	MRS	integer	MODE DECAY RESTRIKE
RTT	integer	READ TRANSFER	NRS	integer	MODE NUMBER RESTRIKE
RCI	integer	READ COUPLE (INNER)	TSPS	real	TIME START PRESTRIKE
RTL	integer	READ TRANSMISSION LINE	TRPS	real	TIME RISE PRESTRIKE
RFT	integer	READ INVERSE	TDPS	real	TIME DECAY PRESTRIKE
WLS	integer	WRITE STROKE	TPPS	real	TIME PULSE PRESTRIKE
WCO	integer	WRITE COUPLE (OUTER)	TCPS	real	TIME CONSTANT PRESTRIKE
WTT	integer	WRITE TRANSFER	TSRX	real	TIME START RETURN
WCI	integer	WRITE COUPLE (INNER)	TRRX	real	TIME RISE RETURN
WTL	integer	WRITE TRANSMISSION LINE	TDRX	real	TIME DECAY RETURN
WFT	integer	WRITE INVERSE	TPRX	real	TIME PULSE RETURN
MR	integer	MODE ROOT	TCRX	real	TIME CONSTANT RETURN

Table 1. (Cont.)

<u>Numeric</u>	<u>Type</u>	<u>Description</u>	<u>Numeric</u>	<u>Type</u>	<u>Description</u>
TSRS	real	TIME START	NW	integer	NUMBER OF SAMPLES
TRRS	real	TIME RISE	ITI	integer	INITIAL TIME
TDRS	real	TIME DECAY	ITF	integer	FINAL TIME
TPRS	real	TIME PULSE	NTI	integer	NUMBER OF SAMPLES
TCRS	real	TIME CONSTANT	RW	real	RADIUS WIRE
SI	real	INITIAL EXCITATION	PSM	real	RADIUS SHIELD MINUS
SF	real	FINAL EXCITATION	RSP	real	RADIUS SHIELD PLUS
NS	integer	NUMBER OF BANDS	RAN	real	RADIUS ARMOR MINUS
XI	real	INITIAL POSITION	RAF	real	RADIUS ARMOR PLUS
XF	real	FINAL POSITION	RS	real	RADIUS SHEATH
NX	integer	NUMBER OF BANDS	DO	real	DEPTH
YI	real	INITIAL POSITION	SO	real	SEPARATION
YF	real	FINAL POSITION	URE	real	RELATIVE PERMEABILITY EARTH
NY	integer	NUMBER OF BANDS	ERE	real	RELATIVE PERMITTIVITY EARTH
WI	real	INITIAL FREQUENCY	OE	real	ABSOLUTE CONDUCTIVITY EARTH
WF	real	FINAL FREQUENCY	URI	real	RELATIVE PERMEABILITY SHEATH
NW	integer	NUMBER OF BANDS	ERI	real	RELATIVE PERMITTIVITY SHEATH
TI	real	INITIAL TIME	OI	real	ABSOLUTE CONDUCTIVITY SHEATH
TF	real	FINAL TIME	URA	real	RELATIVE PERMEABILITY ARMOR
NT	integer	NUMBER OF BANDS	ERA	real	RELATIVE PERMITTIVITY ARMOR
MS	integer	MODE EXCITATION	OA	real	ABSOLUTE CONDUCTIVITY ARMOR
MX	integer	MODE POSITION	UR2	real	RELATIVE PERMEABILITY 2
MY	integer	MODE POSITION	ER2	real	RELATIVE PERMITTIVITY 2
MW	integer	MODE FREQUENCY	O2	real	ABSOLUTE CONDUCTIVITY 2
MT	integer	MODE TIME	URS	real	RELATIVE PERMEABILITY SHIELD
IEI	integer	INITIAL EXCITATION	ERS	real	RELATIVE PERMITTIVITY SHIELD
IEF	integer	FINAL EXCITATION	OS	real	ABSOLUTE CONDUCTIVITY SHIELD
NEI	integer	NUMBER OF EXCITATIONS	URI	real	RELATIVE PERMEABILITY 1
IXI	integer	INITIAL POSITION	ERI	real	RELATIVE PERMITTIVITY 1
IXF	integer	FINAL POSITION	OI	real	ABSOLUTE CONDUCTIVITY 1
NXI	integer	NUMBER OF SAMPLES	URW	real	RELATIVE PERMEABILITY WIRE
IYI	integer	INITIAL POSITION	ERW	real	RELATIVE PERMITTIVITY WIRE
IYF	integer	FINAL POSITION	OW	real	ABSOLUTE CONDUCTIVITY WIRE
NYI	integer	NUMBER OF SAMPLES	ZLM	complex	LOAD IMPEDANCE MINUS
IWI	integer	INITIAL FREQUENCY	ZLP	complex	LOAD IMPEDANCE PLUS
IWF	integer	FINAL FREQUENCY			

Table 1. (Cont.)

Table 2. Peak current and voltage, rise time, and half-life decay time of the pulses on the inner conductor.

current (amps)															voltage (volts)												
d	0 <sub>m</sub>					500 <sub>m</sub>					1000 <sub>m</sub>					0 <sub>m</sub>				500 <sub>m</sub>				1000 <sub>m</sub>			
	I <sub>peak</sub>	t <sub>p</sub>	t <sub>d</sub>	I <sub>peak</sub>	t <sub>p</sub>	t <sub>d</sub>	I <sub>peak</sub>	t <sub>p</sub>	t <sub>d</sub>	I <sub>peak</sub>	t <sub>p</sub>	t <sub>d</sub>	V <sub>peak</sub>	t <sub>p</sub>	t <sub>d</sub>	V <sub>peak</sub>	t <sub>p</sub>	t <sub>d</sub>	V <sub>peak</sub>	t <sub>p</sub>	t <sub>d</sub>	V <sub>peak</sub>	t <sub>p</sub>	t <sub>d</sub>			
0	36.5	25μs	85μs	253	45μs	110μs	13.1	50μs	110μs	128	15μs	45μs	674	30μs	90μs	652	50μs	110μs									
10	36.2	25μs	85μs	251	45μs	110μs	13.0	50μs	110μs	127	15μs	45μs	669	30μs	90μs	647	50μs	110μs									
50	31.3	25μs	85μs	217	45μs	110μs	11.3	50μs	110μs	110	15μs	45μs	579	30μs	90μs	561	50μs	110μs									
100	23.9	25μs	85μs	167	45μs	110μs	8.69	50μs	115μs	84.5	15μs	45μs	443	30μs	95μs	432	50μs	115μs									
200	15.8	25μs	85μs	112	45μs	110μs	5.81	50μs	115μs	56.0	15μs	45μs	295	30μs	95μs	289	50μs	115μs									
500	7.37	25μs	90μs	52.5	45μs	110μs	2.77	45μs	110μs	25.9	15μs	50μs	138	30μs	90μs	137	45μs	110μs									
1,000	3.06	25μs	90μs	22.0	45μs	110μs	1.16	45μs	110μs	10.7	15μs	50μs	57.4	30μs	90μs	57.7	45μs	110μs									
2,000	0.855	25μs	90μs	6.17	45μs	110μs	.327	45μs	110μs	2.96	15μs	50μs	16.0	30μs	70μs	16.2	45μs	110μs									
5,000	0.0848	25μs	90μs	.614	45μs	110μs	0.0325	45μs	110μs	.293	15μs	50μs	1.59	30μs	95μs	1.61	45μs	110μs									
10,000	0.0115	25μs	90μs	.0836	45μs	110μs	0.00443	45μs	110μs	0.0397	15μs	50μs	.216	30μs	95μs	.219	45μs	110μs									